(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 130 514 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
05.09.2001 Bulletin 2001/36

(51) Int Cl.⁷: G06F 11/14

(21) Application number: 00119434.9

(22) Date of filing: 14.09.2000

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 03.03.2000 JP 2000063289

(71) Applicant: Hitachi, Ltd.
Chiyoda-ku, Tokyo 101-8010 (JP)

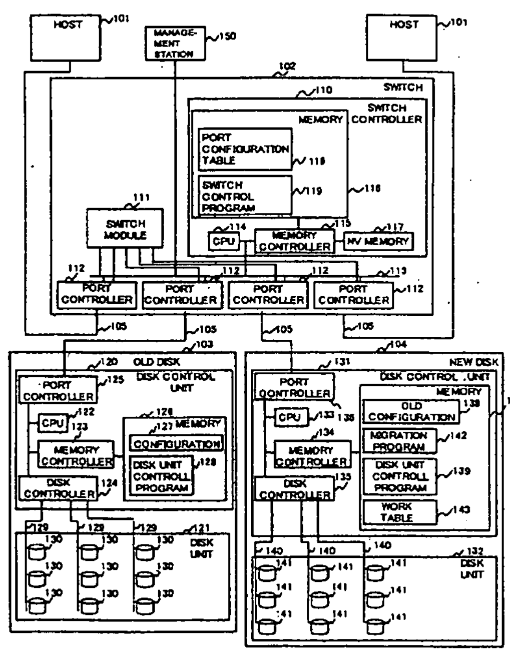
(72) Inventors:
- Watanabe, Naoki, Int. Prop. Gp., Hitachi, Ltd. Chiyoda-ku, Tokyo 100-8220 (JP)
- Takamoto, Yoshifumi, Int. Prop. Gp., Hitachi, Ltd. Chiyoda-ku, Tokyo 100-8220 (JP)

(74) Representative: Beetz & Partner Patentanwälte
Steinsdorfstrasse 10
80538 München (DE)

(54) Data migration method using storage area network

(57) The invention relates to a method of migrating data, wherein a new disk system is connected to a switch that has been already connected to a host (101) and old disk system (103). At that time, the new disk system is assigned in advance a port ID of the same value as the port ID that has been assigned to the F_Port of the switch, so that the new disk system will not be recognized from host (101). The new disk system read the configuration information of the old disk system. Then, in the switch, physical port ID (602) of old disk system (103) and physical port ID (602) of new disk system (104) are exchanged. The new disk system constructs volumes corresponding to the logical volume numbers and sizes of those volumes in the old disk system, and stores data of the old disk system in the corresponding volumes.

Fig 1



EP 1 130 514 A2

Description

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

[0001] The present invention relates to a method of additionally attaching a new disk system (migration destination disk system) to a storage area network (hereafter referred to as SAN) and migrating data from an old disk system (migration source disk system), which is already connected to the SAN, to that new disk system and a device thereof.

DESCRIPTION OF RELATED ART

[0002] Conventional computer systems are configured with a plurality of servers connected by a network such as a LAN and disk units are directly connected to each computer. The data inside each disk unit was managed by the directly connected server. In other words, data was managed in a distributed condition.

[0003] However, the SAN which adopts a configuration in which a plurality of servers, a plurality of disk units, backup units, etc. are connected by a switch or hub has recently become a hot topic. Fiber channels are used to physically connect these units to a switch or hub. The advantages of configuring a system using a SAN are superior scalability and the realization of reduced management cost since unified management can be performed by integrating data distributed over a plurality of disk units. Consequently, a SAN is well suited to a large-scale system. Because many resources are interconnected with a SAN, the management of these resources is an important topic. The management of a logical volume (such as backup), as the management of a disk unit, is one example of that. At present, these management methods are being studied by the SNIA (Storage Networking Industry Association) and NSIC (National Storage Industry Consortium).

[0004] On the other hand, US patent no. 5,680,640 discloses a system and method of migrating data from a first data storage system to a second data storage system. A summary of that is as follows. A first data storage system already connected to a host, network or other data processing system is detached from the aforementioned host and connected to a second data storage system. The aforementioned second data storage system is connected to the aforementioned host or aforementioned data processing system. The aforementioned second data storage system contains a data map or data table that indicates which data element is stored in the aforementioned second storage system and which data in the aforementioned first data storage system has been copied to the aforementioned second storage system. When the aforementioned host, aforementioned network or aforementioned data processing system issues a request for data to the aforementioned second storage system, the aforementioned second data storage system judges whether that data has been stored in the aforementioned second data storage system or in the aforementioned first storage system. If the data has been stored in the aforementioned second data storage system, that data can be used by the host, etc. If the data is not stored in the aforementioned second data storage system, the aforementioned second data storage system issues a request to the aforementioned first data storage system, and that data can be used by the aforementioned host, etc. Then, the aforementioned second data storage system writes that data to its own system and updates the aforementioned data map or aforementioned data table. When the aforementioned second data storage system is not busy, it scans the aforementioned data map or aforementioned data table, determines which data of the aforementioned first data storage system has not yet been copied into its own system, and executes the copying of that data and updating of the aforementioned data map or aforementioned data table.

SUMMARY OF THE INVENTION

[0005] According to the data migration system and method disclosed in US patent no. 5,680,640, independent from the aforementioned host, it is possible to execute data migration from the aforementioned first data storage system to the aforementioned second data storage system..

[0006] However, in the aforementioned data migration system and method, when the aforementioned second data storage system is utilized, the procedure of detaching the aforementioned first data storage system from the aforementioned host, connecting it to the aforementioned second data storage system, and connecting the aforementioned second data storage system to the aforementioned host is necessary. Consequently, at least during the interval beginning when the aforementioned first data storage system is detached from the aforementioned host until the aforementioned second data storage system is connected to the aforementioned host and the aforementioned first storage system, the aforementioned host cannot issue I/O (Input/Output) requests. In addition, during that interval it is necessary to temporarily suspend applications or other programs on the aforementioned host that use the aforementioned first data storage system. To further suppress costs associated with data migration, it is necessary to further reduce the time when I/O requests cannot be issued and the time when applications are suspended.

[0007] In addition, in the aforementioned data migration system and method, to connect the aforementioned first data storage system to the aforementioned second data storage system, a channel must be newly provided to each data storage system.

[0008] Further, in US patent no. 5,680,640, a data migration system and method in a SAN environment are

not disclosed.

[0009] The object of the present invention is to provide data migration systems and methods that are well suited to a SAN environment.

[0010] An overview of the typical data migration of the present invention is as follows. A host computer and first disk unit are each connected by means of a communication channel to a switch port, and it is assumed that the aforementioned first disk unit receives read/write requests from the aforementioned host computer via the aforementioned switch. Each port of the aforementioned switch is assigned a physical port ID and a logical port ID. The aforementioned switch is provided with a table that maintains the correspondence relation between the physical port ID and the logical port ID. The second disk unit is connected to a port of the aforementioned switch that is different from the port connected to the aforementioned host computer and the port connected to the aforementioned first disk unit. Via the aforementioned switch, the aforementioned second disk unit obtains configuration information (the number of logical volumes and size of each logical volume, for example) of the aforementioned first disk unit. Then, the correspondence relation between the physical port ID and the logical port ID assigned to the port of the switch connected to the aforementioned first disk unit is exchanged with the correspondence relation between the physical port ID and the logical port ID assigned to the port of the switch connected to the aforementioned second disk unit. Concretely, the logical port ID assigned to the port of the switch connected to the aforementioned first disk unit and the logical port ID assigned to the port of the switch connected to the aforementioned second disk unit are exchanged. Consequently, even if the host computer attempts to access the aforementioned first disk unit, in actuality, the aforementioned second disk unit will be accessed. After this port ID switching process is performed, the aforementioned second disk unit is configured with the logical volume that corresponds to the configuration information of the aforementioned first disk, and data in the aforementioned first disk unit is migrated to the aforementioned second disk unit. If there is a read or write request from the aforementioned host computer for data already migrated to the aforementioned second disk unit, that processing is performed for that data by the aforementioned second disk unit. If there is a read or write request from the aforementioned host computer for data that has not yet migrated to the aforementioned second disk unit, that data is read from the aforementioned first disk unit into the aforementioned second disk unit and the processing is performed for that data by the aforementioned second disk unit.

[0011] Other data migration methods provided by this application are clarified with the preferred embodiments of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] FIG. 1 is a diagram that describes the configuration of the computer system in the first preferred embodiment of the present invention.

[0013] FIG. 2 is a diagram that describes the configuration of the host.

[0014] FIG. 3 is a diagram that describes the fabric configured with one switch.

[0015] FIG. 4 is a diagram that describes the fabric configured with a plurality of switches.

[0016] FIG. 5 is a diagram that describes the frame of the fiber channel.

[0017] FIG. 6 indicate an example configuration of the port configuration table of the present invention: FIG. 6 (A) indicates an example configuration of the port configuration before port switching and FIG. 6 (B) indicates an example configuration of the port configuration after port switching.

[0018] FIG. 7 indicates an example configuration of the old configuration information of the present invention.

[0019] FIG. 8 indicates an example structure of the migration worktable of the present invention.

[0020] FIG. 9 is a flowchart of the data migration procedure of the present invention.

[0021] FIG. 10 is a flowchart of the port switching procedure of the present invention.

[0022] FIG. 11 is a flowchart of the data migration procedure of the present invention.

[0023] FIG. 12 is a diagram that describes the configuration of the computer system in the second preferred embodiment of the present invention.

[0024] FIG. 13 is a diagram that describes the configuration of the computer system in the third preferred embodiment of the present invention.

[0025] FIG. 14 indicates an example configuration of the port configuration table of the third preferred embodiment of the present invention.

[0026] FIG. 15 indicates a flowchart of the data migration procedure of the third preferred embodiment of the present invention.

[0027] FIG. 16 is a diagram that describes the configuration of the computer system in the fourth preferred embodiment of the present invention.

[0028] FIG. 17 indicates an example configuration of the migrator of the present invention.

[0029] FIG. 18 is a diagram that describes the configuration of the computer system in the fifth preferred embodiment of the present invention.

[0030] FIG. 19 is a diagram that describes the configuration of the host in the fifth preferred embodiment of the present invention.

[0031] FIG. 20 is a diagram that describes the correspondence between the physical port ID and logical port ID assigned to each port prior to the port switching procedure of the present invention.

[0032] FIG. 21 is a diagram that describes the corre-

spondence between the physical port ID and logical port ID assigned to each port after the port switching procedure of the present invention.

[0033] FIG. 22 is an example configuration of the case where the management station of the present invention is connected to a LAN.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0034] Below, preferred embodiments 1 through 5 propose data migration methods and devices, well suited to a SAN environment.

Preferred Embodiment 1

[0035] In the present preferred embodiment, a new disk system is connected to a switch that is already connected to a host and an old disk system, and using that switch, data on the old disk system is migrated to the new disk system. Therefore, because with the present preferred embodiment, there is no need to perform the task of detaching the connection between the host and old disk system and connecting a new disk system to the host and old disk system, and it is only required to connect a new disk system to the switch, compared to the data migration system and method disclosed in the aforementioned US patent no. 5,680,640, the work of adding a new disk system is reduced. Thus, the costs associated with data migration can also be suppressed. In addition, since it is unnecessary to provide a dedicated channel for data migration with the present preferred embodiment, hardware costs can also be reduced.

[0036] Below, the present preferred embodiment will be described using FIG. 1 through FIG. 10.

(1) System Configuration

[0037] FIG. 1 is a diagram that describes the first preferred embodiment of a computer system to which the data migration system and method of the present invention is applied. Two hosts 101, old disk system 103 and new disk system 104 are connected with switch 102, and these form a SAN. Hosts 101, old disk system 103 and new disk system 104 are connected to switch 102 with fiber channel 105. Since the present invention performs data migration from old disk system 103 to new disk system 104, there is no limitation on the number of hosts.

[0038] FIG. 5 indicates the frame format that is used with fiber channel 105. Between SOF (Start Of Frame) 501 that indicates the beginning of the frame and EOF (End Of Frame) 505 that indicates the end, frame header 502, data field 503 and CRC (Cyclic Redundancy Check) 504 are inserted into the frame. Frame header Control) 506 that controls the routing, TYPE 510 that indicates the data structure, F_CTL (Frame Control) 511 that controls the frame sequence and exchange, SEQ_ID (Sequence_ID) 512 that distinguishes between the sending source and receive side sequences, SEQ_CNT (Sequence count) 514 that indicates the count value of a number of frames of each sequence, and data field control information DF_CNTL (Data Field Control) 513. Details of the frame of fiber channel 105 are shown in ANSI X3.230 FC-PH (Fiber Channel Physical and Signaling Interface).

[0039] FIG. 3 indicates a logical network connection configuration of the computer system shown in FIG. 1. In this computer system, fabric 301 is configured with one switch 102. Port 303 of switch 102, used in the connection of host 101 and the disk units, is called the F (Fabric) port. Also, ports 302 of host 101 and the disk units are called N (Node) ports. As indicated in FIG. 4, fabric 301 can also be configured with a plurality of switches 102. The switch ports that connect to other switches are called the E (Expansion) ports 401.

[0040] In the case where the fabric is configured with a single switch 102 as in FIG. 3, data transfers from host 101 to new disk system 104 are performed as follows. Host 101 stores its own port ID in S_ID 508, stores the port ID of a frame's sending destination in D_ID 507, and then sends the frame. When switch 102 receives the frame, it examines the port ID which has been stored in D_ID 507 and sends the frame from the F_Port which matches that port ID to the N_Port of new disk system 104. S_ID 508 and D_ID are logical port IDs.

[0041] Further, as shown in FIG. 4, in the case where fabric 301 is configured from a plurality of switches 102, data transfers from host 101 to new disk system 104 are performed as follows. When switch 102 that is directly connected to host 101 receives a frame from host 101, it examines the port ID which has been stored in D_ID 507. However, because that switch does not have the F_Port which matches that port ID, it sends the frame from an E_Port to another switch. Next, the switch that has the F_Port which matches that port ID and is directly connected to new disk system 104 sends the frame from that F_Port to the N_Port of new disk system 104. The data transfers from host 101 to new disk system 104 have been described above as an example, but data transfers are performed in the same manner among other devices.

[0042] Again, referencing FIG. 1, the configurations of switch 102, old disk system 103, new disk system 104 and host 101 will be described in this order.

(A) Switch 102 Configuration

[0043] Switch 102 includes switch controller 110, switch module 111 and port controller 112

[0045] CPU 114, using internal bus 113, performs the exchange of control information and data with each controller.

[0046] The read/write access to nonvolatile memory 117 is controlled by memory controller 115, and it stores such data as switch control program 119 which is necessary for the control of switch 102 and port configuration table 118 which is necessary during execution of that program.

[0047] FIG. 6 indicates an example configuration of port configuration table 118. FIG. 6 (A) shows a port configuration table before port switching, and FIG. 6 (B) shows a port configuration table after port switching. The port switching procedure will be described later.

[0048] Port configuration table 118 indicates the port configuration, and includes logical port ID 601 that indicates the logical port ID and physical port ID 602 that indicates the physical port ID. In addition, it may also include the rule of the transfer service class of the fiber channel and the port type such as loop. In the present preferred embodiment, as indicated in FIG. 20, it is assumed that a host 101 is connected to logical port ID_0, old disk system 103 is connected to logical port ID_1, and a host 101 is connected to logical port ID_3. Also, it is assumed that new disk system 104 is connected to logical port ID_2.

[0049] The read/write access to memory 116 is controlled by memory controller 115. When switch control program 119 is executed by CPU 114, that program is read from nonvolatile memory 117 and stored in memory 116. Further, at that time and as necessary, CPU 114 also reads port configuration table 118 from nonvolatile memory 117 and stores it in memory 116.

[0050] Port controller 112 performs control of the data coding/compounding. Concretely, port controller 112 extracts serial data from a received fiber channel frame and converts it into parallel data, and extracts information necessary for switching such as the send destination ID from the frame and sends it to switch module 111. Also, port controller 112 performs the reverse procedure.

[0051] Switch module 111 has a plurality of ports that are individually connected to a plurality of port controllers 112. Switch module 111 is connected to CPU 114 via internal bus 113 and is controlled by CPU 114. When switch module 111 receives data from port controller 112, in accordance with the send destination ID of that data, it switches a port to the output port and sends data. Switch module 111 may be made with a crossbar switch.

[0053] In FIG. 1, management station 150 is directly connected to switch 102. However, as shown in FIG. 22, the aforementioned settings may be implemented with the Web or other means, by connecting host 101, switch 102, old disk system 103, new disk system 104 and management station 150 to a LAN (Local Area Network).

(B) Old disk system 103 and New disk system 104 Configurations

[0054] In present preferred embodiment, in order to simplify the description, old disk system 103 and new disk system 104 are assumed to have the same configuration, except for programs and information in memory.

[0055] Old disk system 103 includes disk control unit 120 and disk unit 121.

[0056] Disk unit 121 includes a plurality of disk drives 130. These disk drives 130 are connected to disk controller 124 with interface (I/F) 129 (fiber channel, SCSI, etc.). In the implementation of the present invention, the number of disk drives 130 is not restricted.

[0057] Disk controller 120 includes CPU 122, memory controller 123, port controller 125, disk controller 124 and memory 126. CPU 122, memory controller 123, port controller 125 and disk controller 124 are connected with an internal bus.

[0058] CPU 122, using the internal bus, performs the exchange of control information and data with these controllers. CPU 122 issues read and write commands to disk drive 130 that is necessary in the processing of commands sent from host 101 via switch 102. CPU 122 constructs the well-known RAID 0~5 configuration with a plurality of disk drives 130, and provides logical volumes for host 101.

[0059] Port controller 125 is connected to port controller 112 of switch 102 by fiber channel 105 and has the same functions as port controller 112.

[0060] Memory 126 is connected to memory controller 123 and stores disk unit control program 128 that controls the disk unit, data that are necessary during execution of said program, and configuration information 127. Configuration information 127 will be described later. Disk unit control program 128 is executed by CPU 122 and is a program that controls port controller 125 and disk controller 124, and processes read/write commands received from host 101.

[0061] As described above, except for the program and information in memory 137, the configuration ele-

[0062] FIG. 8 indicates an example structure of migration worktable 143. Migration worktable 143 includes volume number 801, slot number 802 and status 803 that indicates the status of said slot.

(C) Host 101 Configuration

[0063] FIG. 2 indicates the configuration of host 101. Host 101 includes CPU 201, memory controller 202, port controller 206, disk controller 204, disk drive 205 and memory 203.

[0064] CPU 201, memory controller 202, port controller 206 and disk controller 204 are connected with internal bus 207. CPU 201, using internal bus 207, performs the exchange of control information and data with these controllers.

[0065] Port controller 206 is connected to port controller 112 of switch 102 with fiber channel 105 and performs the exchange of commands and data with port controller 112.

[0066] Disk drive 205 is connected to disk controller 204 and stores operating system 208, device driver 209 that controls hardware such as port controller 206, application program 210 and data that is necessary during execution of these programs.

[0067] Memory 203 is connected to memory controller 202. When operating system 208, disk controller 204, device driver 209, application program 210, etc. are executed by CPU 201, they are read from disk drive 205 and stored in memory 203.

(2) Data Migration Procedure Flow

[0068] Next, referencing FIG. 9, the data migration procedure of the present invention will be described. It is assumed that host 101 and old disk system 103 have been already connected to switch 102.

(A) Connect New Disk System to Switch (901)

[0069] First, an operator activates management station 150 and then inputs the port numbers of switch 102 that are connected to old disk system 103 and new disk system 104.

[0070] Next, the operator connects new disk system 104 to switch 102. At this point in time, if the port ID of new disk system 104 is a port ID other than a port ID already in use, any port ID may be assigned. When each port of switch 102 is assigned a port ID by default, the port ID assigned to the F_Port connected to new disk system 104 is appropriate. In the present preferred embodiment, as described above, logical port ID_2 is assigned to new disk system 104.

[0071] Because the logical port ID assigned to new disk system 104 is different from logical port ID_1 of the disk unit that can be used by host 101, at this point in time, host 101 cannot access new disk system 104.

(B) Transfer Configuration of Old disk system to New disk system (902)

[0072] After connecting new disk system 104, the operator activates migration program 142 on new disk system 104 with the aforementioned command from management station 150. The activated migration program 142 first obtains configuration information 127 from old disk system 103 (902).

[0073] As indicated in FIG. 7, configuration information 127 includes old port ID 702, world wide name 703, fiber configuration information 704 and SCSI configuration information 705.

[0074] Fiber configuration information 704 includes PLOGI payload 706 and PRLI payload 707.

[0075] PLOGI payload 706 includes common service parameter 708 that are exchanged at the time of a fiber channel N_Port login (PLOGI), port name 709, node name 710 and class service parameter 711. Specified in common service parameter 708 are the fiber channel version information and functions such as the address specification method supported by said fiber channel device, the communication method, etc. Class service parameter 711 indicates the class support information, X_ID re-assignment, ACK capability, etc. Details of fiber channel parameters are described in the aforementioned ANSI X3.230 FC-PH. PRLI payload 707 includes service parameter 712.

[0076] SCSI configuration information 705 includes inquiry data 703, disconnect/reconnect parameter 714, sense data 715 and mode select parameter 716. Inquiry data 713 indicates the type, vendor ID, product ID, etc. of the SCSI device. Disconnect/reconnect parameter 714 indicates the connection conditions. When an error occurs, sense data 715 is exchanged to investigate the state of the disk unit. Mode select parameter 716 performs the setting and modification of various parameters related to the physical attributes of the SCSI device, data format of the storage media, error recovery method and procedure, processing method of an I/O process, etc. By means of inquiry data 713 and mode select parameter 716, the number of volumes and size of each volume (number of blocks) on old disk system 103 can be learned. A detailed description of the protocol for SCSI on fiber channel 105 is shown in ANSI X.269 Fiber Channel Protocol for SCSI.

[0077] As described above, configuration information 127 can also exist as information in addition to that which can be acquired using protocols of the existing fiber channel, SCSI, etc. If that type of configuration information exists, the operator directly reads configuration information 127 from old disk system 103 and transfers it to new disk system 104. Reading and transferring are performed from an operation panel or by web access by means of HTTP.

[0078] When the transfer of configuration information 127 to new disk system 104 is completed, or in other words, when the transfer of old configuration information

138 is completed, migration program 142 of new disk system 104 notifies switch 102 of that fact.

(C) Port Switching Procedure (903)

[0079] When switch control program 119 of switch 102 receives that notification, the port switching procedure is started.

[0080] Details of the port switching procedure are described using the flowchart of FIG. 10.

[0081] Initially, the case as indicated in FIG. 3 where fabric 301 is configured with a single switch 102 will be described.

[0082] First, switch control program 119 of switch 102 notifies all hosts 101 using old disk system 103 that port switching has begun. Device driver 209 of host 101 that has received this notification queues an I/O process for old disk system 103 in memory 203 of host 101 (1001).

[0083] When an I/O process for old disk system 103 is suspended, device driver 209 notifies switch 102 of the I/O suspension completion. If an I/O process for old disk system 103 is being executed, it may be terminated, but it is desirable to allow the execution to finish and then issue notification of the I/O suspension completion. After receiving such notification from all hosts 101, switch control program 119 of switch 102 changes the correspondence relation between logical port ID 601 and physical port ID 602 of old disk system 103 and the correspondence relation between logical port ID 601 and physical port ID 602 of new disk system 104 of port configuration table 118 (1002). In other words, port configuration table 118 is rewritten as in FIG. 6 (B). Using diagrams of the whole computer system, this situation can be illustrated as the change from the state indicated in FIG. 20 to the state indicated in FIG. 21.

[0084] Thereafter, by means of referencing port configuration table 118 at each frame transmission or reception, port controller 112 of switch 102 performs the port switching procedure by manipulating S_ID 508 and D_ID 507. When a frame is received, logical port ID 601 that corresponds to S_ID 508 of the frame is retrieved, and S_ID 508 of the frame is converted into physical port ID 602 that corresponds to the retrieved logical port ID 601. Similarly, when a frame is transmitted, logical port ID 601 that corresponds to D_ID 507 of the frame is retrieved, and that D_ID 507 is converted into physical port ID 602 that corresponds to the retrieved logical port ID 601. At this time, CRC 504 that is appended to the frame is recalculated. By means of the above procedure, all frames to old disk system 103 are sent to new disk system 104. Further, frames sent from new disk system 104 are viewed by host 101 as being sent from old disk system 103. After port switching by switch 102, host 101 I/O is restarted (1003).

[0085] Next, as indicated in FIG. 4, the case where fabric 301 is configured with a plurality of switches 102 will be described.

[0086] First, switch 102 which will become the master (master switch) is determined. In the present preferred embodiment, the master switch is switch 102 that is directly connected to new disk system 104.

[0087] Switch control program 119 of the master switch notifies all hosts 101 that are using old disk system 103 and all switches except for the master switch in fabric 301 of the starting of port switching. The processing to be performed by device driver 209 of host 101 that has received such notification is the same as in the case where the fabric is configured from a single switch 102, except for notification to the master switch of the I/O suspension completion.

[0088] After such notification is received from all hosts 101, switch control program 119 of the master switch changes the correspondence relation between logical port ID 601 and physical port ID 602 of old disk system 103 and the correspondence relation between logical port ID 601 and physical port ID 602 of new disk system 104 of port configuration table 118, and notifies all switches 102 except the master switch of that change. Based on such notification, all switches except for the master switch change their own port configuration table 118. Thereafter, the operation of each switch is the same as the operation of switch 102 in the case where the fabric is configured with a single switch 102. After port switching has been performed by all switches 102 in fabric 301, host 101 I/O is restarted.

(D) Data Migration Procedure (904)

[0089] The data migration procedure is performed synchronously with the completion of the port switching procedure. This data migration procedure will be described using the flowchart of FIG. 11.

[0090] First, migration program 142 constructs a volume in new disk system 104 that corresponds to the logical volume number and size of that volume in old disk system 103, and then initializes variables to be used and migration worktable 143 that was described with FIG. 8 (1101).

[0091] Migration program 142 on new disk system 104 checks whether there is an I/O request from host 101 (1102).

[0092] If there is no I/O request from host 101, data is migrated in slot units from old disk system 103. At that time, using migration worktable 143, migration program 142 computes the address of the next data to migrate (1103).

[0093] Data migration is performed for all the logical volumes of old disk system 103 in order of increasing volume number 801. Further, within each volume, data migration is performed beginning with the first block. Worktable 143 for migration shown in FIG. 8 indicates the next migration address in the volume number 801 cell of the first row. The initial value of the next migration address is the first block address of the smallest volume number 801. Moreover, migration worktable 143 indicates the next migration slot in the slot number 802 cell

of the first row. From the second row onward, the rows are arranged in order of increasing volume number 801 and slot number 802. When migration program 142 transfers data of the next migration address obtained from migration worktable 143, it changes status 803 of that slot number to "Migrating" (1104).

[0094] Next, using port controller 125, migration program 142 issues a request to read the slot size to old disk system 103 and acquires the corresponding data (1105).

[0095] Migration program 142 writes the data obtained from old disk system 103 to a corresponding volume of new disk system 104 on disk drive 141, using disk controller 124 and via I/F 140 (1106).

[0096] After the data is written on disk drive 141, it is checked whether the migration of all slots of the corresponding volume is completed. If completed, data migration of the next volume is executed. If not completed, the slot number is incremented (1107).

[0097] If the migration of all volumes is completed, the migration procedure is finished (1108).

[0098] Next, the case of an I/O request from host 101 will be described. If there is an I/O request from host 101, migration program 142 investigates whether the request is for a read or write access (1109).

[0099] In the case of a read request, migration program 142 references migration worktable 143 and investigates whether migration of the requested data has been completed (1110).

[0100] If the migration is completed, migration program 142 reads that data from disk drive 141 in new disk system 104 and sends it back to host 101 (1113, 1112).

[0101] If the migration of the data requested from host 101 is not complete, using port controller 125, migration program 142 issues a read request to old disk system 103 via switch 102 and acquires said data. Migration program 142 sends back the data acquired from old disk system 103 to host 101 and also writes that data to the corresponding volume of new disk system 104 (1111, 1112).

[0102] After said data is written on disk drive 141, said data is registered into migration worktable 143, and with status 803 set to "Done", the processing of said request is finished (1108).

[0103] Next, the case of a write request from host 101 will be described. Migration program 142 receives data related to the write request from host 101 (1114).

[0104] Then, using port controller 125, migration program 142 issues a read request in slot size units to old disk system 103 via switch 102 so as to include the data requested from host 101, and acquires the data. After migration program 142 overwrites the data obtained from the migration source with the data received from host 101 and creates new slot data, it writes to the volume in new disk system 104 requested by host 101 (1115).

[0105] After the write is completed, migration program 142 registers said slot into migration worktable 143, and with status 803 set to "Done", the processing related to the write request from host 101 is finished (1107).

(E) Detach Old disk system (905)

[0106] After the data migration procedure is completed, the operator detaches old disk system 103 from the switch and terminates the migration procedure (905).

[0107] The system configuration and flow of the data migration procedure have been described above for preferred embodiment 1. By means of the present preferred embodiment, I/O requests from the host are only suspended during the time while configuration information of old disk system 103 is being transferred to new disk system 104 and during the time while the port switching procedure is being performed. Therefore, by means of the present preferred embodiment, because there is no need to perform the task of connecting a new disk system to the host and old disk system, by detaching the connection between the host and old disk system, the time while I/O requests from the host are suspended is shorter than the time while I/O requests from the host are suspended in the data migration system and method disclosed in US patent no. 5,680,640. Consequently, the costs associated with data migration can be suppressed. Moreover, by means of the present preferred embodiment, since there is no need to provide a dedicated channel for data migration, hardware costs can also be suppressed.

Preferred Embodiment 2

[0108] The present preferred embodiment differs from preferred embodiment 1 in that in addition to the system configuration described with preferred embodiment 1, the old disk system and new disk system are directly connected with an interface (fiber channel or SCSI, for example). In the present preferred embodiment, since data migration is performed using that interface, compared to preferred embodiment 1, high-speed data migration can be performed. In the present preferred embodiment, it is necessary to provide an interface for data migration. However, there is no need to perform the task of connecting a new disk system to the host and old disk system, by detaching the connection between the host and old disk system, and because it is only required to connect a new disk system to the switch and connect the old disk system and the new disk system with an interface, compared to the data migration system and method disclosed in the aforementioned US patent no. 5,680,640, the work of adding the new disk system is reduced.

[0109] Below, the present embodiment will be described using FIG. 12.

[0110] FIG. 12 is a diagram that describes the second preferred embodiment of a computer system that has adopted the data migration system and method of the present invention. As described above, each configura-

tion element of the computer system is the same as each configuration element of FIG. 1, and therefore their descriptions are omitted.

[0111] Below, the flow of the data migration procedure in the present preferred embodiment will be described. The data migration procedure flow is the same as that of preferred embodiment 1, which was described with FIG. 9. However, in the present preferred embodiment, since old disk system 1203 and new disk system 1204 are connected directly by a interface and data migration is performed using that interface, it is necessary to obtain the correspondence relation between the logical address and physical address in old disk system 1203 from old configuration information 1238. Below, only this point of difference will be described.

[0112] In steps 1105, 1113 and 1115 of FIG. 11, data is read from the old disk system via switch 102. However, in the present preferred embodiment, using information that is included in old configuration information 1238, migration program 1231 computes the physical address corresponding to the logical address of the data that should be read from old disk system 1203. Then, using I/F 1240, migration program 1231 issues directly a read request to disk drive 1230 of old disk system 1203 that corresponds to the computed physical address, and acquires that data. At this time, if that data has been distributed to a plurality of disk drives 1230 of old disk system 1203, or if stored in dispersed physical addresses, a plurality of read requests will be issued.

[0113] In the present preferred embodiment, since old disk system 1203 and new disk system 1204 are connected directly with an interface, and data migration is performed using that interface without passing through switch 1202, compared to preferred embodiment 1, high-speed data migration can be performed. Further, since the data migration of the present preferred embodiment does not pass through switch 1202, the resource consumption of the switch can be suppressed. In the present preferred embodiment, two hosts and two disk units are connected to switch 1202. However, because many hosts and disk units are connected in a SAN environment, those resources can be allocated to the data transfer between other hosts and other disk units. Therefore, by means of the present preferred embodiment, compared to preferred embodiment 1, the degradation of switch performance that accompanies data migration can be suppressed.

Preferred Embodiment 3

[0114] The present preferred embodiment differs from preferred embodiment 1 in that the migration program is executed by a switch. By means of the present preferred embodiment, since it is unnecessary for a disk unit to have a migration program, data migration from an old disk system is possible even with a disk unit that does not have this type of program.

[0115] Using FIG. 13, the configuration of the computer system of the present preferred embodiment will be described. However, a description of those configuration elements that are the same as configuration elements of FIG. 1 will be omitted.

[0116] In the present preferred embodiment, non-volatile memory 1317 in switch 1302 stores switch control program 1319 that is necessary for control of switch 1302, migration program 1342 that performs the migration procedure, port configuration table 1318 that is necessary during execution of these programs, old configuration information 1327 and migration worktable 1343. Disk unit control program 1339 is stored in memory 1337 in new disk system 1304, but the old configuration information, migration worktable, and migration program are not stored, though they are in preferred embodiment 1.

[0117] FIG. 14 is an example configuration of port configuration table 1318 that indicates the port configuration. Port configuration table 1318 includes logical port ID 1401 that indicates the virtual port ID, physical port ID 1402 that indicates the physical port ID, and status 1403 that indicates the port status.

[0118] Next, using the flowchart of FIG. 15, the data migration procedure of the present preferred embodiment will be described.

[0119] An operator connects new disk system 1304 to switch 1302 (1501). This step is the same as step 901 of FIG. 9.

[0120] Next, migration program 1342 of switch 1302 reads configuration information 1327 from old disk system 1303 (1502). Configuration information 1327 is the same as that shown in FIG. 7.

[0121] After configuration information 1327 of old disk system 1303 has been transferred to switch 1302, switch control program 1319 of switch 1302 starts the port switching procedure (1503).

[0122] Also in the port switching procedure of the present preferred embodiment, similar to preferred embodiment 1, after the host I/O is stopped, the correspondence relation between logical port ID 1401 and physical port ID 1402 of old disk system 1303 and the correspondence relation between logical port ID 1401 and physical port ID 1402 of new disk system 1304 of port configuration table 1318 are changed. However, the port switching procedure of the present preferred embodiment differs from that of preferred embodiment 1 in the following point. In the present preferred embodiment, switch control program 1319 of switch 1302 sets port ID status 1403 of old disk system 1303 in port configuration table 1318 to "Migrating." The frame for the port ID with "Migrating" status 1403 in port configuration table 1318 is not sent to the corresponding port, but instead is passed to migration program 1331 of switch 1302. In other words, access to old disk system 1303 is performed by migration program 1331 of switch 1302.

[0123] In the case where the fabric is configured with a plurality of switches 1302, using the same method as preferred embodiment 1, after the host I/O is stopped, switch control program 1319 of the master switch sets

port ID status 1403 of old disk system 1303 in port configuration table 1318 to "Migrating." The frame for the port ID with "Migrating" status 1403 in port configuration table 1318 is not sent to the corresponding port, but instead is passed to migration program 1331 of switch 1302. In the case where the fabric is configured with a plurality of switches 1302, it is sufficient if at least old configuration information 1327, migration worktable 1343, and migration program 1342 are provided in switch 1302, the master switch.

[0124] After the port switching procedure is completed, the data migration procedure is performed. First, migration program 1342 of switch 1302 constructs a volume according to the logical volume number and size of said volume which are configured in old disk system 1303, and then initializes variables to be used and migration worktable 1332 that is shown with FIG. 8. The procedure thereafter is the same procedure as described in preferred embodiment 1, except that it is not performed by the migration program of the new disk system, but instead by migration program 1302 of switch 1302.

[0125] After the data migration procedure is completed, switch control program 1319 of switch 1302 changes the status of new disk system 1304 in port configuration table 1318 to "Normal" (1505). The operator detaches old disk system 1303 from switch 1302 and terminates the migration procedure (1506).

Preferred Embodiment 4

[0126] The characteristic of the present preferred embodiment is that a migration program is executed on a migrator configured with a personal computer, workstation, etc. which are connected to a switch. By means of the present preferred embodiment, similar to preferred embodiment 3, since it is unnecessary for a new disk system to have a migration program, data migration from an old disk system is possible even with a disk unit that does not have this type of program. Further, in the present preferred embodiment, as in preferred embodiment 3, since the migration program is not executed on a switch, a load of the switch can be reduced.

[0127] Using FIG. 16, the configuration of the computer system of the present preferred embodiment will be described. However, a description of those configuration elements that are the same as configuration elements of FIG. 1 will be omitted.

[0128] A characteristic of the present preferred embodiment, as will be described later, is that migrator 1606 internally stores the migration program that performs the migration procedure, old configuration information and migration worktable. Memory 1637 in new disk system 1604 stores disk unit control program 1639, but does not store the old configuration information, migration worktable and migration program, though in preferred embodiment 1. Further, different from preferred embodiment 3, in the present preferred embodiment, the migration program, old configuration information and migration worktable are not stored in switch 1602.

[0129] FIG. 17 indicates an example configuration ofmigrator 1606. Migrator 1606 includes CPU 1701, memory controller 1702, port controller 1706, disk controller 1704, disk drive 1705 and memory 1703. CPU 1701 is connected to memory controller 1702, port controller 1706 and disk controller 1704 with internal bus 1701. CPU 1701 performs the exchange of control information and data with each controller. Port controller 1706 is connected to port controller 1612 of switch 1602 with fiber channel 1605 and performs the exchange of commands and data with switch 1602. The port of migrator 1606 is called the N port. The port on the switch 1602 side is called the F port. Disk drive 1705 is connected to disk controller 1704 and receives read and write requests from CPU 1701. Disk drive 1705 stores programs necessary for migrator 1606, and stores migrator program 1731 and migrator worktable 1732 that is necessary during execution of said program. Memory 1703 is connected to memory controller 1702. During execution of migration program 1731, this program is read from drive 1705 into memory 1703.

[0130] Next, the flow of the data migration procedure of the present preferred embodiment will be described. An operator connects new disk system 1604 to switch 1602 for data migration. At this time, the port ID of new disk system 1604 takes the same value as the port ID assigned to the F_Port of switch 1602, and new disk system 1604 is not recognized from host 1601. This step is the same as step 901 described in FIG. 9.

[0131] Next, migration program 1631 of migrator 1606 reads configuration information 1727 from old disk system 1603. Configuration information 1727 is the same as that indicated in FIG. 7.

[0132] After configuration information 1727 of old disk system 1603 is transferred to the migrator, the port switching procedure is performed. This procedure, similar to that described with preferred embodiment 1, suspends I/O from the host, and exchanges physical port ID 1604 of old disk system 1603 and physical port ID 1402 of new disk system 1604 in port configuration table 1618. The control program of switch 1602 sets the port ID status of old disk system 1603 in port configuration table 1618 to "Migrating." If the status in the port configuration table is "Migrating", the frame is not sent to a port, but instead is passed to migration program 1731 of migrator 1606. Access to old disk system 1603 is performed by migration program 1731 ofmigrator 1606. After these procedures, the I/O of host 1601 is restarted and migration program 1731 of migrator 1606 starts the migration procedure.

[0133] After the port switching procedure is completed, the data migration procedure is performed. First, migration program 1731 of migrator 1606 constructs a volume according to the logical volume number and size of said volume which are configured in old disk system 1603, and then initializes variables to be used and mi-

gration worktable 1732. The procedure thereafter is the same procedure as described in preferred embodiment 1, except that it is not performed by the migration program of the new disk system, but instead by migration program 1731 of migrator 1606.

[0134] After the data migration procedure is completed, switch control program 1619 of switch 1602 changes the status of new disk system 1604 in port configuration table 1618 to "Normal," as in preferred embodiment 3. The operator detaches old disk system 1603 from switch 1602 and terminates the migration procedure.

[0135] In addition, in the present preferred embodiment, a fiber channel connects the migrator and switch. However, it can be easily understood by those skilled in the art of the present invention that the effect of the present preferred embodiment can be obtained with connection by such means as a vendor-unique bus.

Preferred Embodiment 5

[0136] The characteristic of the present preferred embodiment is that the host implements the port switching of the switch. By means of the present preferred embodiment, there is no need to provide a port configuration table in the switch, though in preferred embodiment 1 through preferred embodiment 4. Therefore, it is possible to perform data migration even in cases where a SAN is configured using a switch that does not have this function.

[0137] Using FIG. 18, an example configuration of the computer system of the present preferred embodiment will be described. However, a description of those configuration elements that are the same as configuration elements of FIG. 1 will be omitted. As will be described later, in the present preferred embodiment, differing from preferred embodiment 1, a port configuration table is provided in host 1801 and a port configuration table is not provided in switch 1802.

[0138] FIG. 19 indicates an example configuration of host 1801. Host 1801 comprises CPU 1901, memory controller 1902, port controller 1906, disk controller 1904, drive 1905 and memory 1903. CPU 1901 is connected to memory controller 1902, port controller 1906 and disk controller 1904 with internal bus 1907. CPU 1901 performs the exchange of control information and data with each controller. Port controller 1906 is connected to port controller 1812 of switch 1802 with fiber channel 1805, and performs the exchange of commands and data with switch 1802. The port of host 1801 is called the N port, and the port on the switch 1802 side is called the F port. Drive 1905 is connected to disk controller 1904 and receives read and write commands from CPU 1901. Drive 1905 stores programs necessary for host 1801, and stores operating system 1908, device driver 1909, application program 1910 and port configuration table 1918. Memory 1903 is connected to memory controller 1902 and receives read and write requests from CPU 1901. Programs such as operating system 1908, disk controller 1904, device driver 1909 that controls the hardware, and application 1910 are read from disk drive 1905 into memory 1903 at the time of each program execution.

[0139] Next, the flow of the data migration procedure of the present preferred embodiment will be described. This flow is the same as the flow of the data migration procedure of preferred embodiment 1, which is shown in FIG. 9. However, in the present preferred embodiment, the port switching procedure is different from that of preferred embodimentl. Here, only the port switching procedure will be described.

[0140] The port switching procedure of the present preferred embodiment performs the same operation both in the case where fabric 301 is configured with a single switch, and in the case where it is configured with a plurality of switches. At first, by operator command, device driver 1909 of the host notifies all hosts 1801 using old disk system 1803 of the start of port switching. Device driver 1909 of host 1801 that has received this notification temporarily stores I/O for old disk system 1803 and then queues it in memory 203 of host 1801. On the other hand, an I/O process being executed for old disk system 1803 is executed until completed. When an I/O process for old disk system 1803 is suspended, device driver 1909 notifies switch 1802 of I/O suspension completion. After receiving I/O suspension completion notification from all hosts 1801, device driver 1909 of the host that issued the notification of port switching requests all the hosts to exchange physical port ID 602 of old disk system 1803 and physical port ID 602 of new disk system 1804 in port configuration table 1918. Thereafter, the device driver of the host references port configuration table 1918 at each frame transmission or reception and performs the port switching procedure by manipulating S_ID 508 and D_ID 507. This procedure is the same as the procedure described with preferred embodiment 1.

[0141] Above, preferred embodiment 1 through preferred embodiment 5 of the present invention have been described. Appropriate combined embodiments of these preferred embodiments are possible. For preferred embodiments 3 through 5, an example would be to connect old disk system 1303 and new disk system 1304 with a disk interface for migration, similar to preferred embodiment 2, and perform data migration via this interface.

[0142] As described above, data migration methods and devices well suited to a SAN environment can be provided by means of this invention.

[0143] Although the present invention has been described in connection with the preferred embodiments thereof, many other variations and modification will become apparent to those skilled in the art.

Claims

1. A method of migrating data from a first storage system that is connected to a switch (102) by means of a communication channel and receives read/write requests from a host computer (101) connected to the switch (102) by means of a communication channel, to a second storage system, comprising the steps of:

   connecting the second storage system to the switch (102) by means of a communication channel; and,
   writing, via the switch (102), data stored in the first storage system to the second storage system.

2. A method of migrating data from a first storage system that is connected to a switch (102) by means of a communication channel and receives read/write requests from a host computer (101) connected to the switch (102) by means of a communication channel, to a second storage system, comprising the steps of:

   connecting the second storage system to the switch (102) by means of a communication channel;
   reading the number of logical volumes constructed in the first storage system and their sizes are read into the second storage system via the switch (102);
   constructing the same number and same sizes of logical volumes as logical volumes that have been constructed in the first storage system in the second storage system; and,
   writing, via the switch (102), data stored in the first storage system to the second storage system in volume units.

3. A method of migrating data from a first storage system that is connected to one port of a switch (102) provided with a plurality of ports each assigned its own port identifier by means of a communication channel and receives, via the switch (102), read/write requests from a host computer (101) connected to another port of said switch (102) by means of a communication channel, to a second storage, the method comprising the steps of:

   connecting the second storage system, by means of a communication channel, to a port, of said switch (102), other than the ports connected to the first disk unit and to the host computer (101);
   mutually exchanging the port identifier assigned to the port connected to the first storage system and the port identifier assigned to the port connected to the second storage system;
   writing the data in the first storage system to the second storage system via the switch;
   if, from said host computer (101), there is a read or write request for the first storage system, sending the request to the second storage system; and,
   if the data for the read or write request has been written already to the second storage system, performing read or write processing of the already written data by means of the second storage system, and if the data for the read or write request has not yet been written to the second storage system, writing the data for the read or write request to the second storage system, and performing read or write processing of the written data by means of the second storage system.

4. A data migration method according to claim 3, further comprising the steps of:

   directly connecting the first storage system and the second storage system by means of a communication channel; and,
   without passing through the switch, writing data in the first storage system to the second storage system via the communication channel that directly connects the first storage system and the second storage system.

5. A data migration method of migrating data from a first storage system that is connected to one port of a switch provided with a plurality of ports each assigned its own port identifier by means of a communication channel and receives, via the switch, read/write requests from a host computer connected to another port of said switch by means of a communication channel, to a second storage, the method comprising the steps of:

   connecting the second storage system to a port, of the switch, other than the ports connected to the first storage system and to the host computer, by means of a communication channel;
   mutually exchanging the port identifier assigned to the port connected to the first storage system and the port identifier assigned to the port connected to the second storage system;
   reading information on logical volumes constructed in the first storage system;
   constructing the same number and same sizes of logical volumes as logical volumes that have been constructed in the first storage system in the second storage subsystem;
   writing data stored in the first storage system to the second storage system via the switch;

managing completion of said writing from the first storage system to the second storage system in volume units;
if, from said host computer, there is a read or write request for the first storage system, sending the request to the second storage system; and,
if the data for the read or write request has been written already to the second storage system, performing read or write processing of the already written data by means of the second storage system, and if the data for the read or write request has not yet been written to the second storage system, writing the data for the read or write request to the second storage system, and performing read or write processing of the written data by means of the second storage system.

6. A data migration method according to claim 5. wherein the second storage system reads, via the switch, the logical volume information from the first storage system and manages the completion of said writing to the second storage system in volume units.

7. A data migration method according to claim 5, wherein the switch reads the logical volume information from the first storage system, constructs the same number and same sizes of logical volumes as logical volumes that have been constructed in the first storage system, in the second storage system, and manages the completion of said writing to the second storage system in volume units.

8. A data migration method according to claim 5, wherein an information processing unit connected to the switch reads the logical volume information from the first storage system, constructs the same number and same sizes of logical volumes as logical volumes that have been constructed in the first storage system, in the second storage system, and manages the completion of said writing to the second storage system in volume units.

9. A data migration method of migrating data from a first storage system that is connected to one port of a switch provided with a plurality of ports each assigned its own port identifier by means of a communication channel and receives, via the switch, read/ write requests from a host computer connected to another port of said switch by means of a communication channel, to a second storage, the method comprising the steps of:

connecting the second storage system to a port, of the switch, other than the ports connected to the first disk unit and to the host computer, by means of a communication channel;
switching the destination of a read/write request from the host computer from the first storage system to the second storage system;
writing data in the first storage system to the second storage system via the switch; and,
if the data for a read or write request from the host computer has been written already to the second storage system, performing read or write processing of the already written data by means of the second storage system, and if the data for the read or write request has not yet been written to the second storage system, writing the data for the read or write request to the second storage system, and performing read or write processing of the written data by means of said second storage system.

10. A data migration method according to claim 9, further comprising the steps of:

reading the number of logical volumes constructed in the first storage system and their sizes;
constructing the same number and same sizes of logical volumes as logical volumes that have been constructed in the first storage system in the second storage system; and,
writing, in volume units, data stored in the first storage system to the second storage system via the switch.

11. A data migration method according to claim 10, wherein the second storage system manages the completion of said writing to the second storage system in volume units.

12. A data migration method according to claim 1, wherein the communication channel connecting between the switch and the first storage system, the communication channel connecting between the switch and the second storage system and communication channel connecting between the switch and the host computer are fiber channels.

13. A disk storage system to which data stored in other storage system connected to a switch is migrated via the switch, comprising:

a port for connecting to the switch,
a program for reading data stored in the other storage system via the switch, if data for a read or write request from a host computer that is connected to the switch has been read already, performing read or write processing of the already read data, and if the data for the read or write request has not yet been read, reading the

data for the read or write request from the other storage system.

14. A computer connected to a switch to which a first storage system is connected, comprising:

a port for connecting to the switch,
a program, in case that a second storage system to which data stored in the first storage system is migrated via the switch is connected to the switch, for changing the destination of a read or write request from the first storage system to the second storage system.

EP 1 130 514 A2

Fig 1

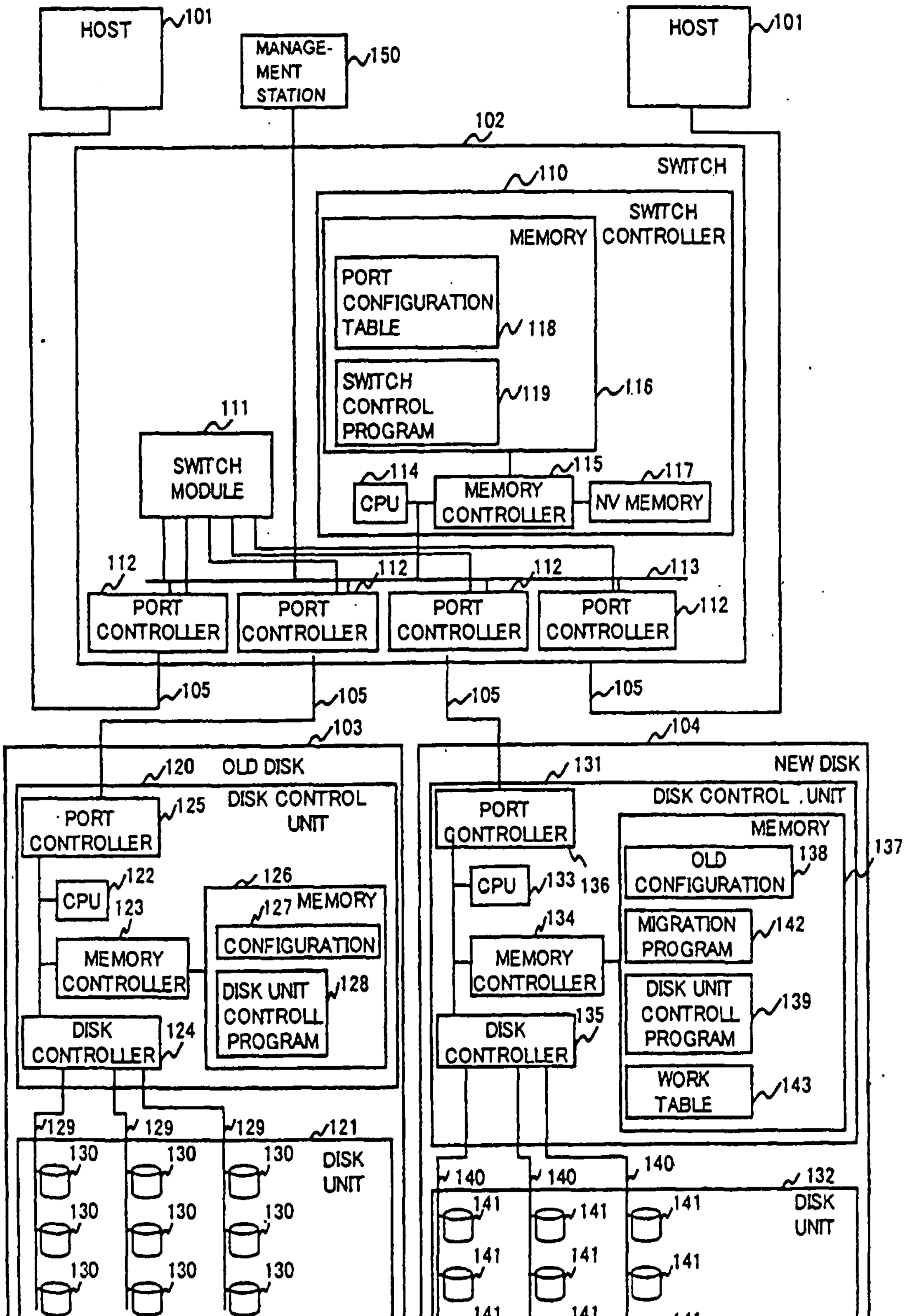


Fig 2

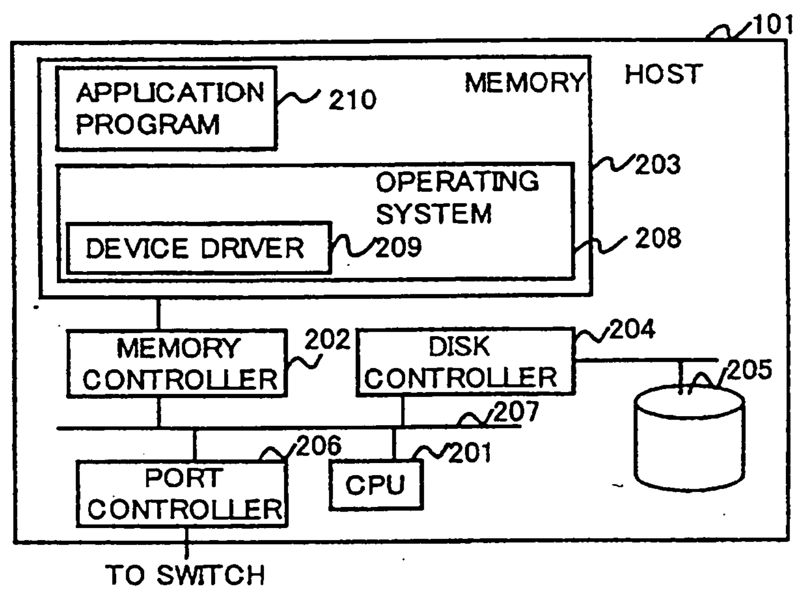
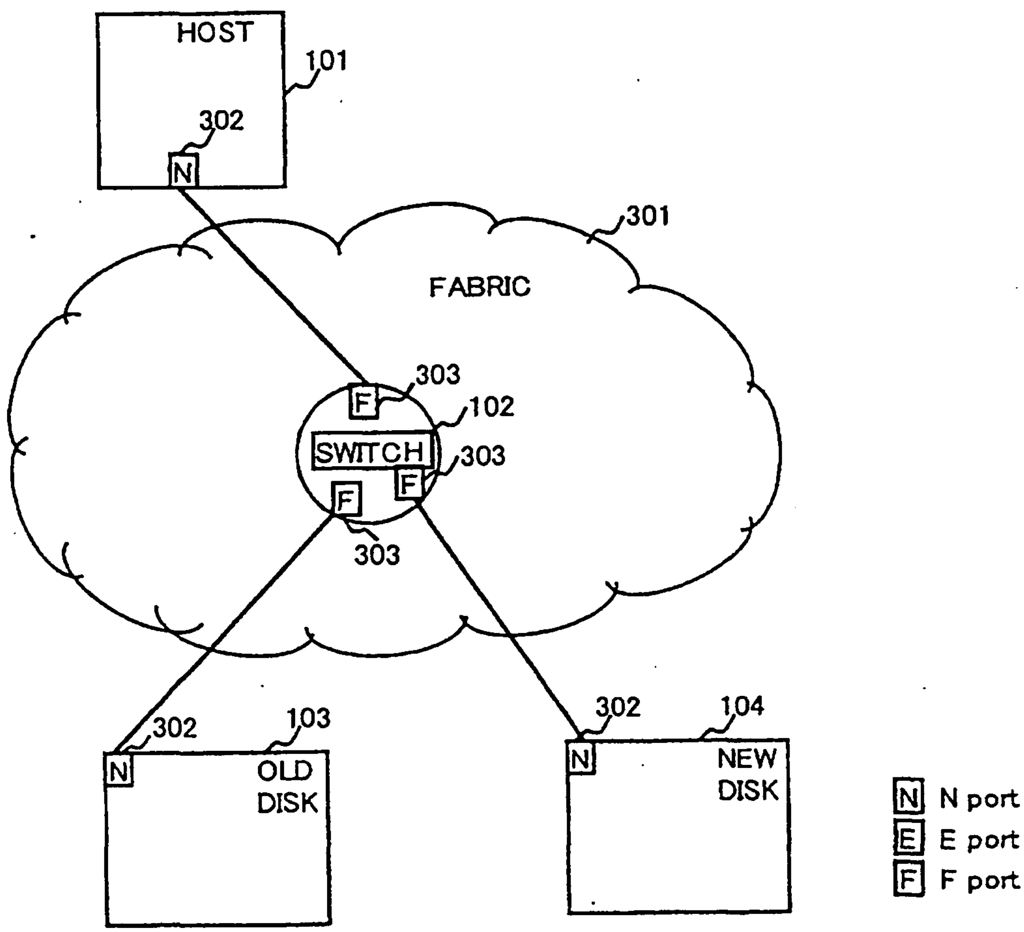

101
HOST
MEMORY
APPLICATION PROGRAM
210
203
OPERATING SYSTEM
DEVICE DRIVER
209
208
MEMORY CONTROLLER
202
DISK CONTROLLER
204
205
207
206
201
PORT CONTROLLER
CPU
TO SWITCH

Fig 3


HOST
101
302
N
301
FABRIC
303
F
102
SWITCH
303
F
F
303
302
103
N
OLD
DISK
302
104
N
NEW
DISK
N N port
E E port
F F port

EP 1 130 514 A2

Fig 4

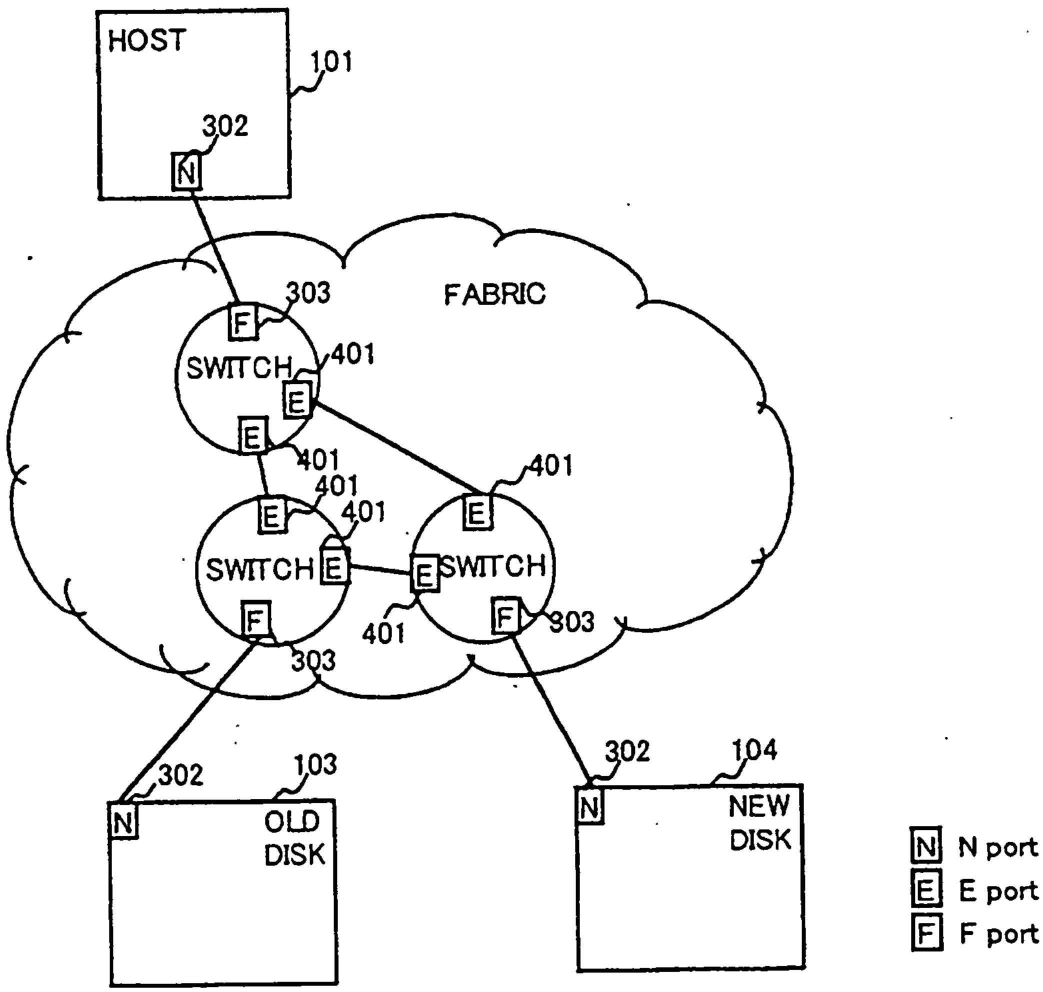


Fig 5

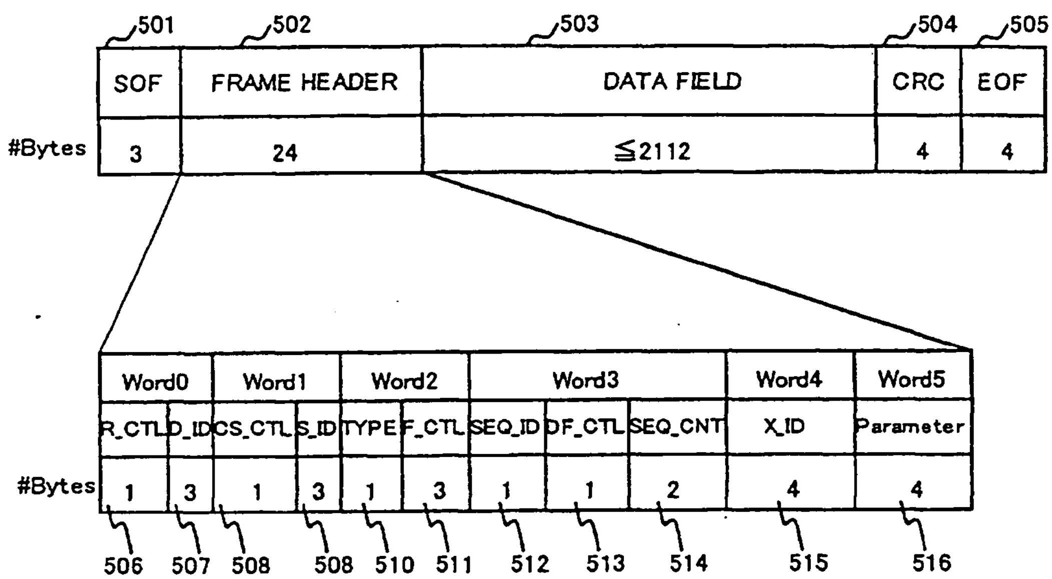

501
502
503
504
505
SOF
FRAME HEADER
DATA FIELD
CRC
EOF
#Bytes
3
24
≦2112
4
4
Word0
Word1
Word2
Word3
Word4
Word5
R_CTL
D_ID
CS_CTL
S_ID
TYPE
F_CTL
SEQ_ID
DF_CTL
SEQ_CNT
X_ID
Parameter
#Bytes
1
3
1
3
1
3
1
1
2
4
4
506
507
508
508
510
511
512
513
514
515
516

Fig 6

601 602

Logical Port ID	Physical Port ID
0	0
1	1
2	2
3	3
⋮	⋮
N-1	N-1

(A) Before Port Switching

601 602

Logical Port ID	Physical Port ID
0	0
1	2
2	1
3	3
⋮	⋮
N-1	N-1

(B) After Port Switching

Fig 7

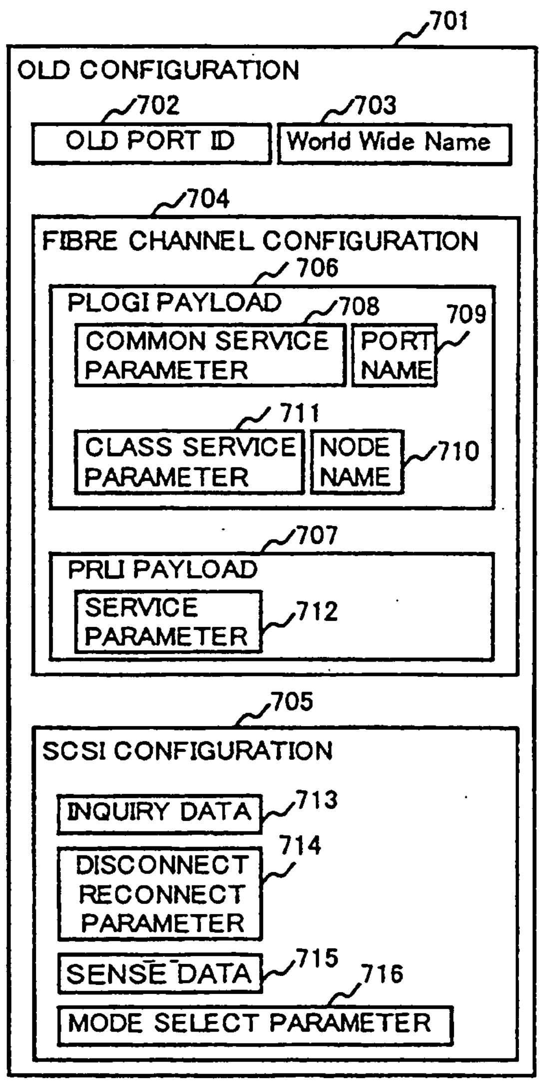

701
OLD CONFIGURATION
702
OLD PORT ID
703
World Wide Name
704
FIBRE CHANNEL CONFIGURATION
706
PLOGI PAYLOAD
708
709
COMMON SERVICE PARAMETER
PORT NAME
711
CLASS SERVICE PARAMETER
NODE NAME
710
707
PRLI PAYLOAD
SERVICE PARAMETER
712
705
SCSI CONFIGURATION
INQUIRY DATA
713
714
DISCONNECT RECONNECT PARAMETER
SENSE DATA
715
716
MODE SELECT PARAMETER

Fig 8

801 802 803

VOLUME NUMBER	SLOT NUMBER	STATUS
1	10	MIGRATING
1	25	DONE
1	26	DONE
2	15	DONE
2	16	DONE
⋮	⋮	⋮

Fig 9

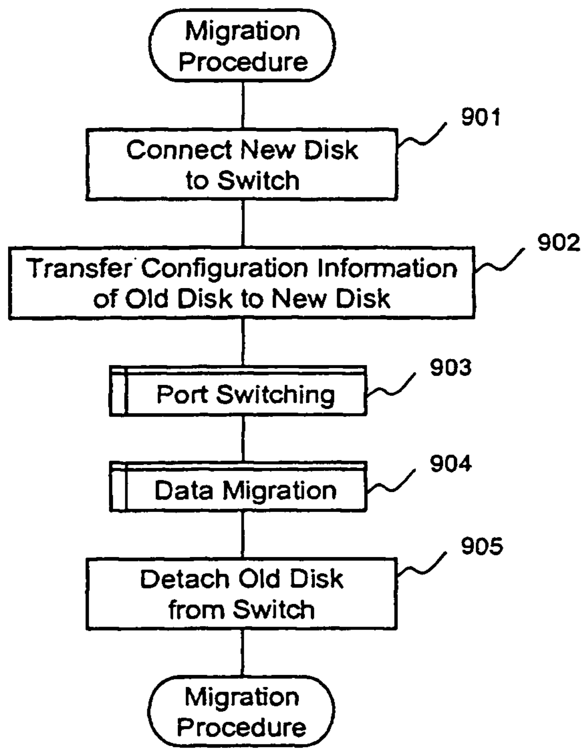

Migration Procedure
Connect New Disk to Switch
901
Transfer Configuration Information of Old Disk to New Disk
902
Port Switching
903
Data Migration
904
Detach Old Disk from Switch
905
Migration Procedure

Fig 10

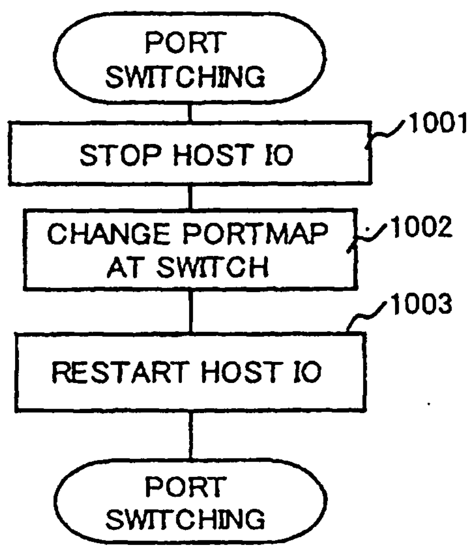
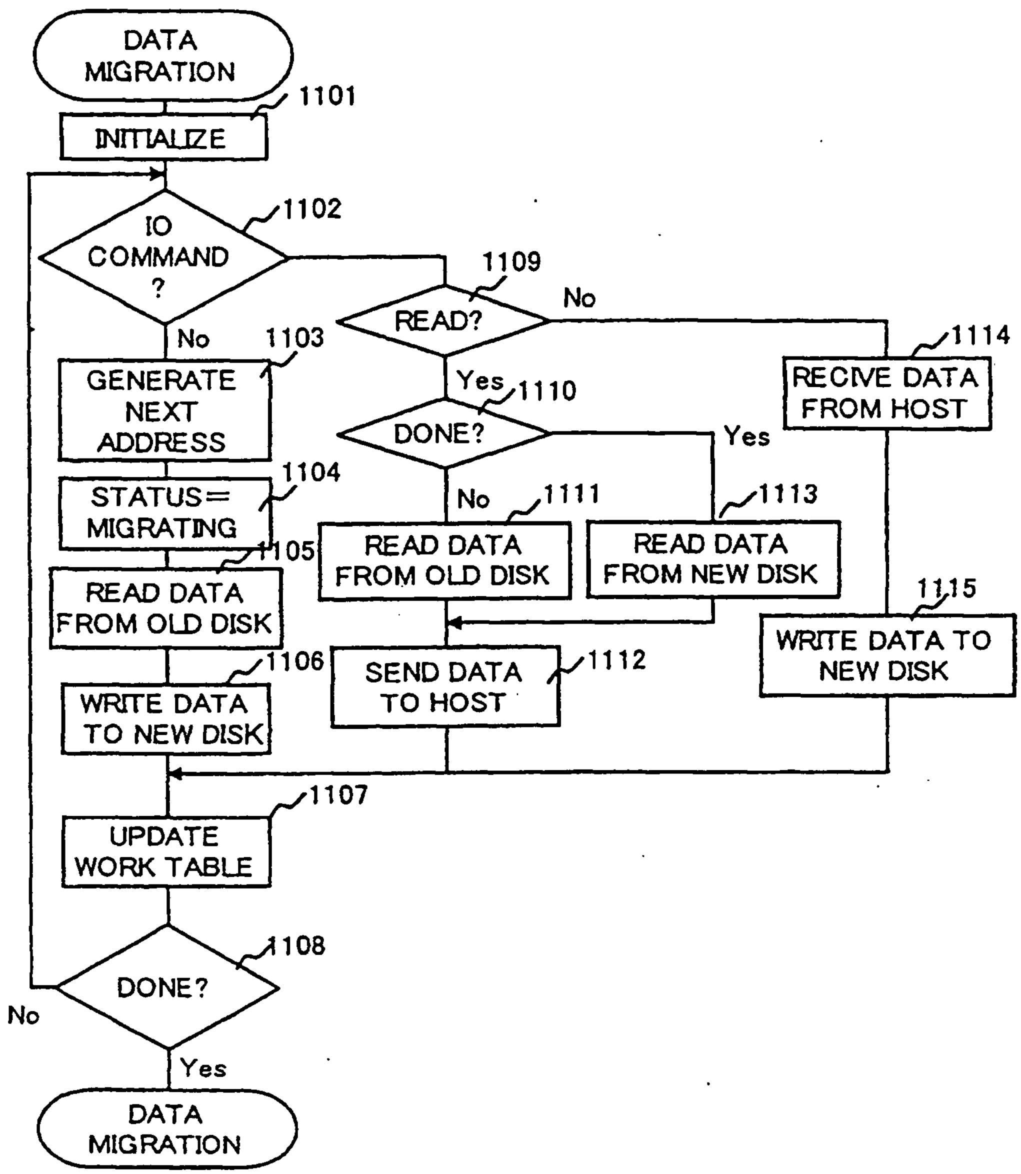

PORT SWITCHING
STOP HOST IO
1001
CHANGE PORTMAP AT SWITCH
1002
1003
RESTART HOST IO
PORT SWITCHING

Fig 11


DATA MIGRATION
1101
INITIALIZE
1102
IO COMMAND ?
No
1103
GENERATE NEXT ADDRESS
1104
STATUS= MIGRATING
1105
READ DATA FROM OLD DISK
1106
WRITE DATA TO NEW DISK
1107
UPDATE WORK TABLE
1108
DONE?
No
Yes
DATA MIGRATION
1109
READ?
No
Yes
1110
DONE?
Yes
No
1111
READ DATA FROM OLD DISK
1113
READ DATA FROM NEW DISK
1112
SEND DATA TO HOST
1114
RECIVE DATA FROM HOST
1115
WRITE DATA TO NEW DISK

EP 1 130 514 A2

Fig 12

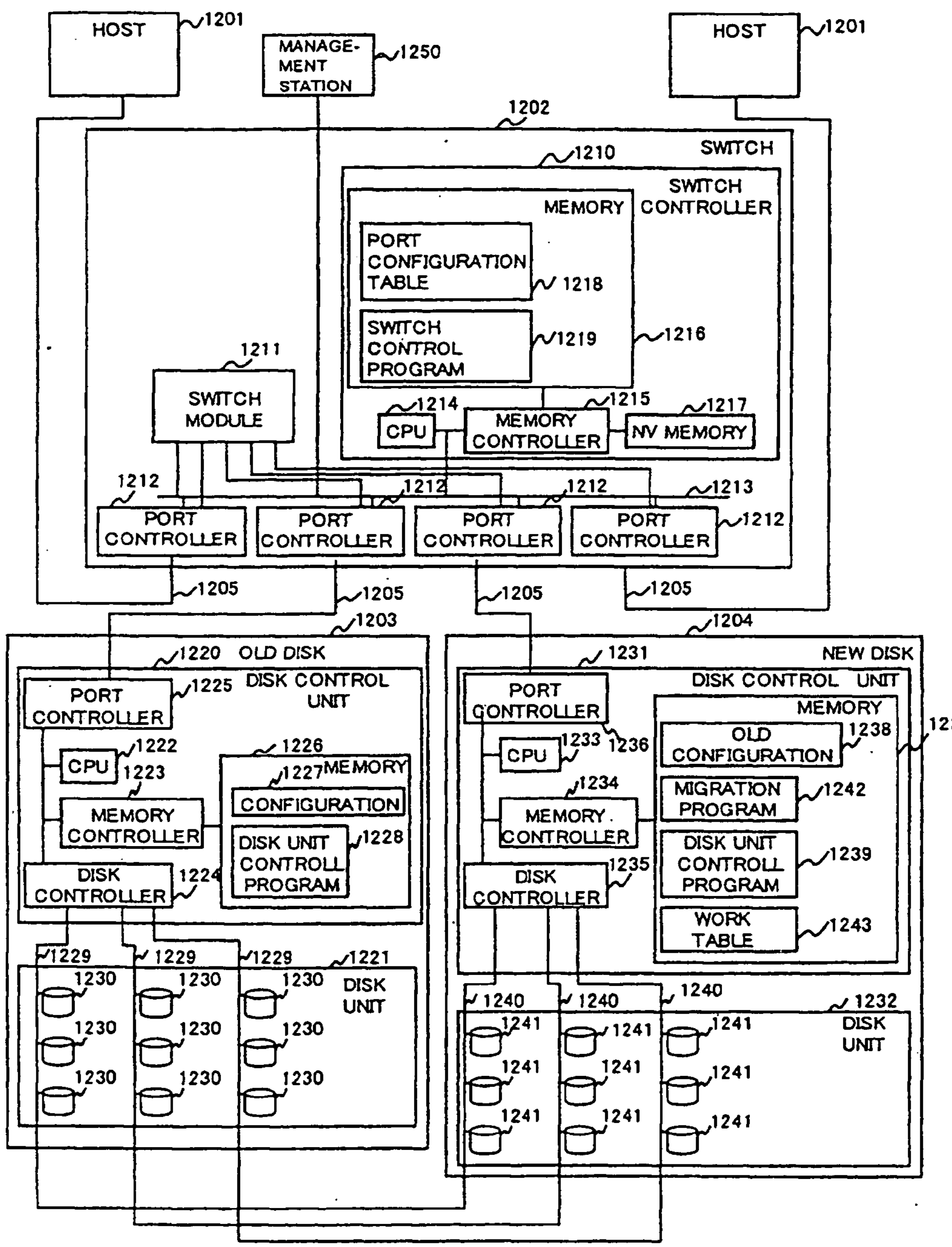


Fig 13

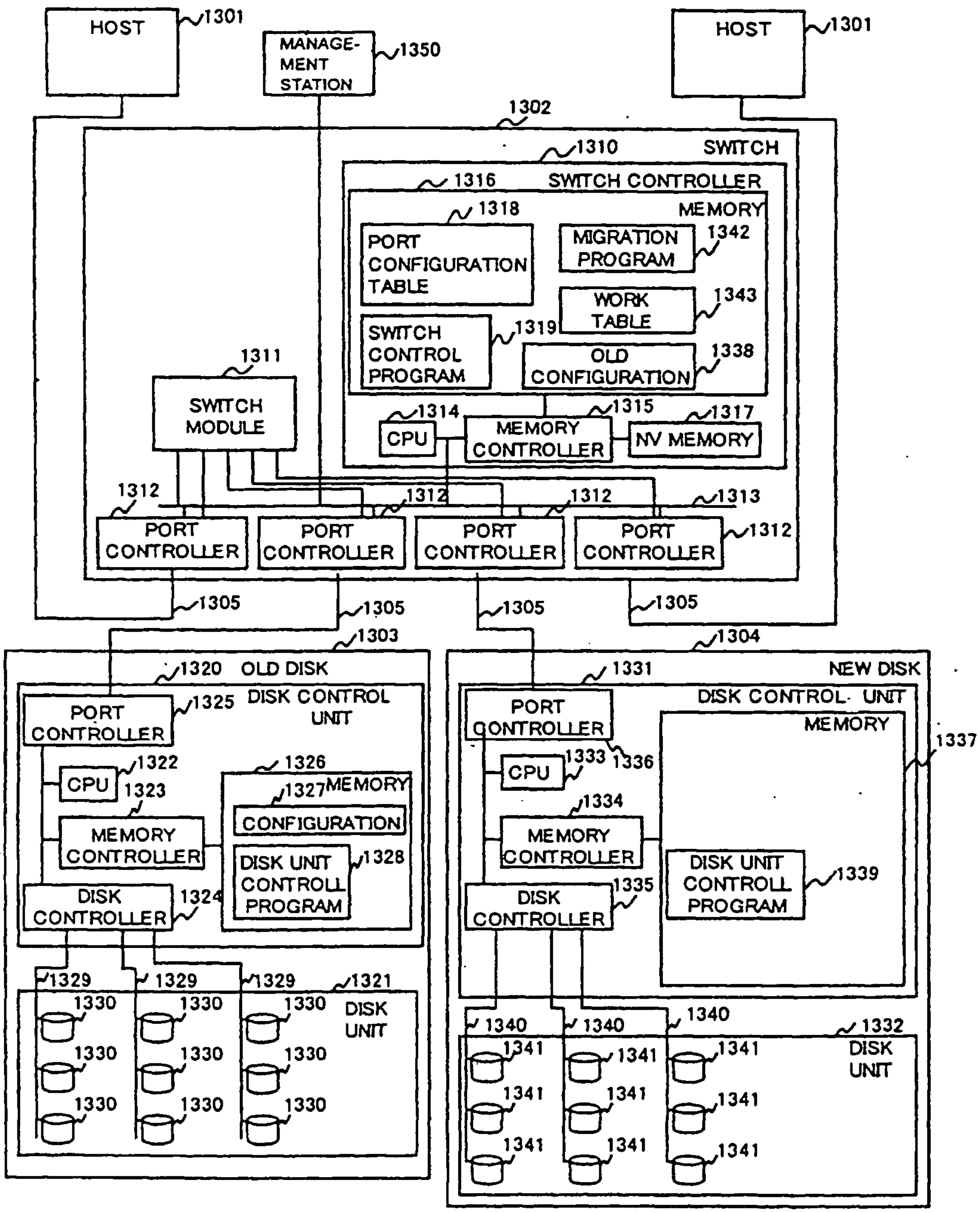

HOST
1301
MANAGE-MENT STATION
1350
HOST
1301
1302
1310
SWITCH
1316
SWITCH CONTROLLER
1318
MEMORY
PORT CONFIGURATION TABLE
MIGRATION PROGRAM
1342
WORK TABLE
1343
SWITCH CONTROL PROGRAM
1319
OLD CONFIGURATION
1338
1311
SWITCH MODULE
1314
CPU
1315
MEMORY CONTROLLER
1317
NV MEMORY
1312
1312
1312
1313
PORT CONTROLLER
PORT CONTROLLER
PORT CONTROLLER
PORT CONTROLLER
1312
1305
1305
1305
1305
1303
1304
1320
OLD DISK
1325
DISK CONTROL UNIT
PORT CONTROLLER
1322
CPU
1323
1326
1327
MEMORY
CONFIGURATION
MEMORY CONTROLLER
DISK UNIT CONTROLL PROGRAM
1328
DISK CONTROLLER
1324
1329
1329
1329
1321
1330
DISK UNIT
1331
NEW DISK
PORT CONTROLLER
DISK CONTROL UNIT
MEMORY
1337
1333
1336
CPU
1334
MEMORY CONTROLLER
DISK UNIT CONTROLL PROGRAM
1339
1335
DISK CONTROLLER
1340
1340
1340
1332
1341
DISK UNIT

Fig 14

1401 1402 1403

Logical Port ID	Physical Port ID	Status
0	0	Normal
1	2	Migrating
2	1	Normal
3	3	Normal
⋮	⋮	⋮
N-1	N-1	Normal

Fig 15

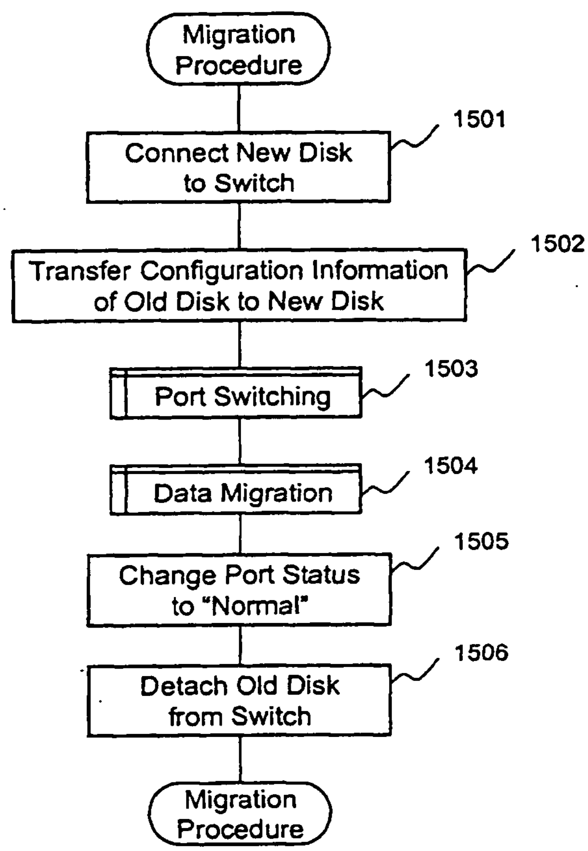
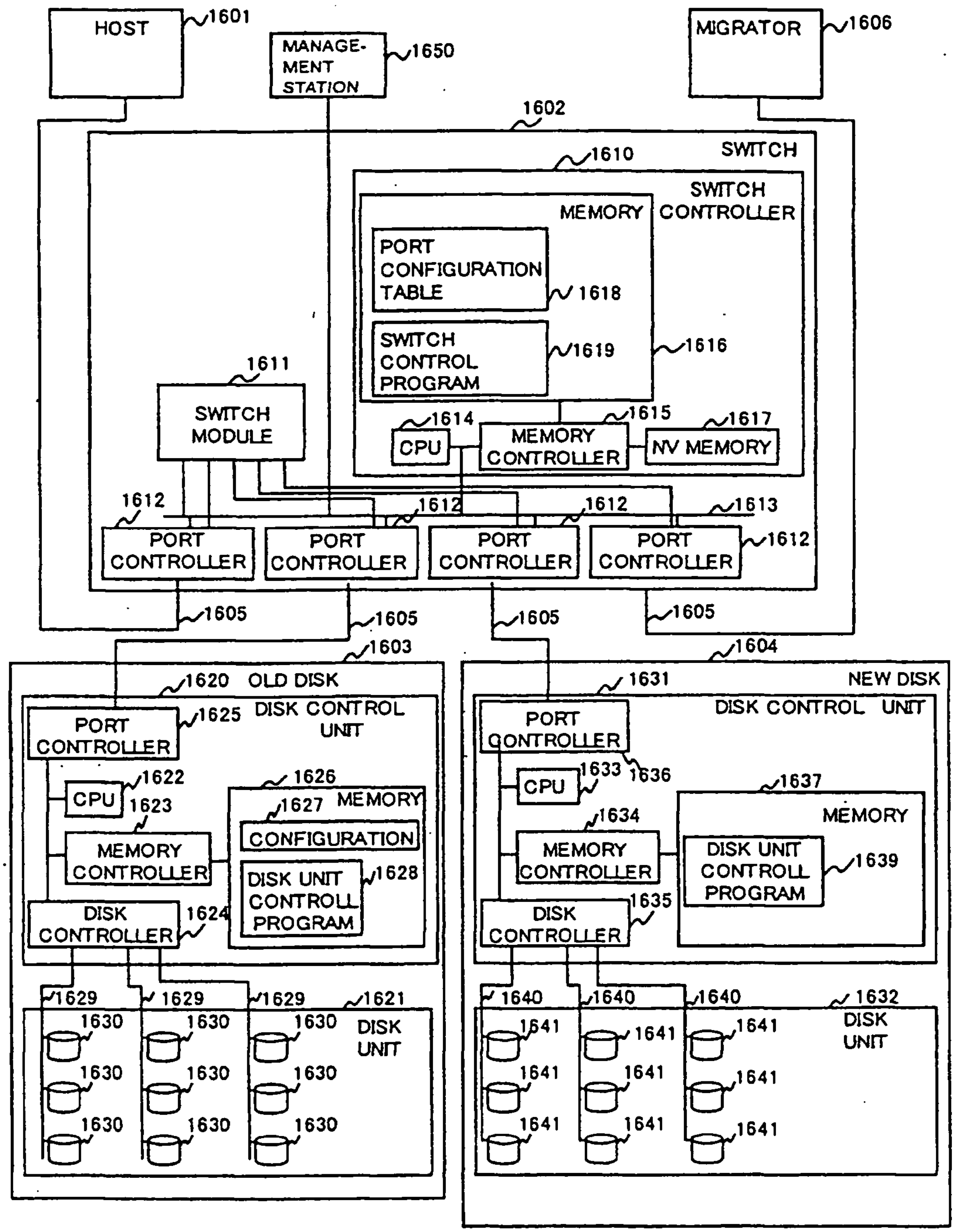
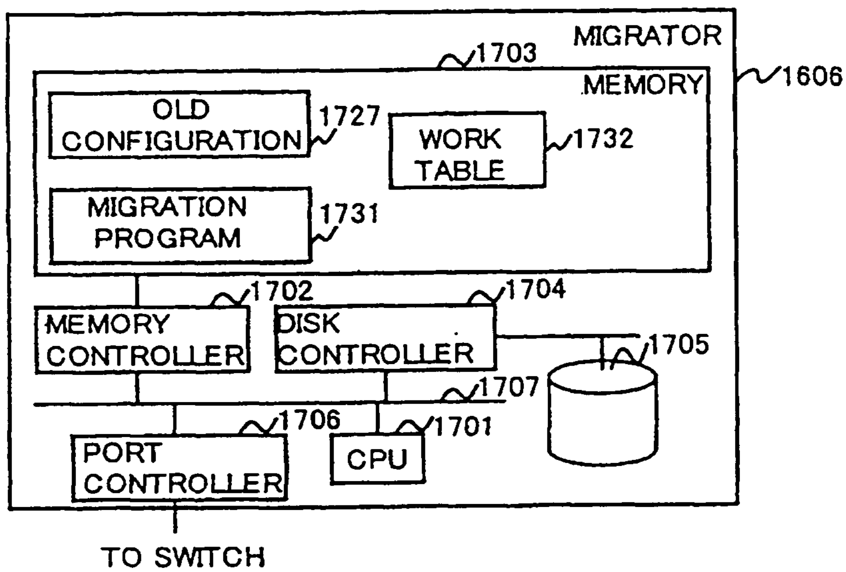

Migration Procedure
Connect New Disk to Switch
1501
Transfer Configuration Information of Old Disk to New Disk
1502
Port Switching
1503
Data Migration
1504
Change Port Status to "Normal"
1505
Detach Old Disk from Switch
1506
Migration Procedure

Fig 16



HOST
1601
MANAGE-MENT STATION
1650
MIGRATOR
1606
1602
1610
SWITCH
SWITCH CONTROLLER
MEMORY
PORT CONFIGURATION TABLE
1618
SWITCH CONTROL PROGRAM
1619
1616
1611
SWITCH MODULE
1614
CPU
1615
MEMORY CONTROLLER
1617
NV MEMORY
1612
1613
PORT CONTROLLER
PORT CONTROLLER
PORT CONTROLLER
PORT CONTROLLER
1605
1603
1604
1620
OLD DISK
1625
DISK CONTROL UNIT
PORT CONTROLLER
CPU
1622
1623
1626
1627
MEMORY
CONFIGURATION
MEMORY CONTROLLER
DISK UNIT CONTROLL PROGRAM
1628
DISK CONTROLLER
1624
1629
1621
1630
DISK UNIT
1631
NEW DISK
DISK CONTROL UNIT
PORT CONTROLLER
1633
1636
CPU
1637
1634
MEMORY
MEMORY CONTROLLER
DISK UNIT CONTROLL PROGRAM
1639
1635
DISK CONTROLLER
1640
1632
1641
DISK UNIT

Fig 17


MIGRATOR
1703
MEMORY
1606
OLD CONFIGURATION
1727
WORK TABLE
1732
MIGRATION PROGRAM
1731
1702
1704
MEMORY CONTROLLER
DISK CONTROLLER
1705
1707
1706
1701
PORT CONTROLLER
CPU
TO SWITCH

EP 1 130 514 A2

Fig 18

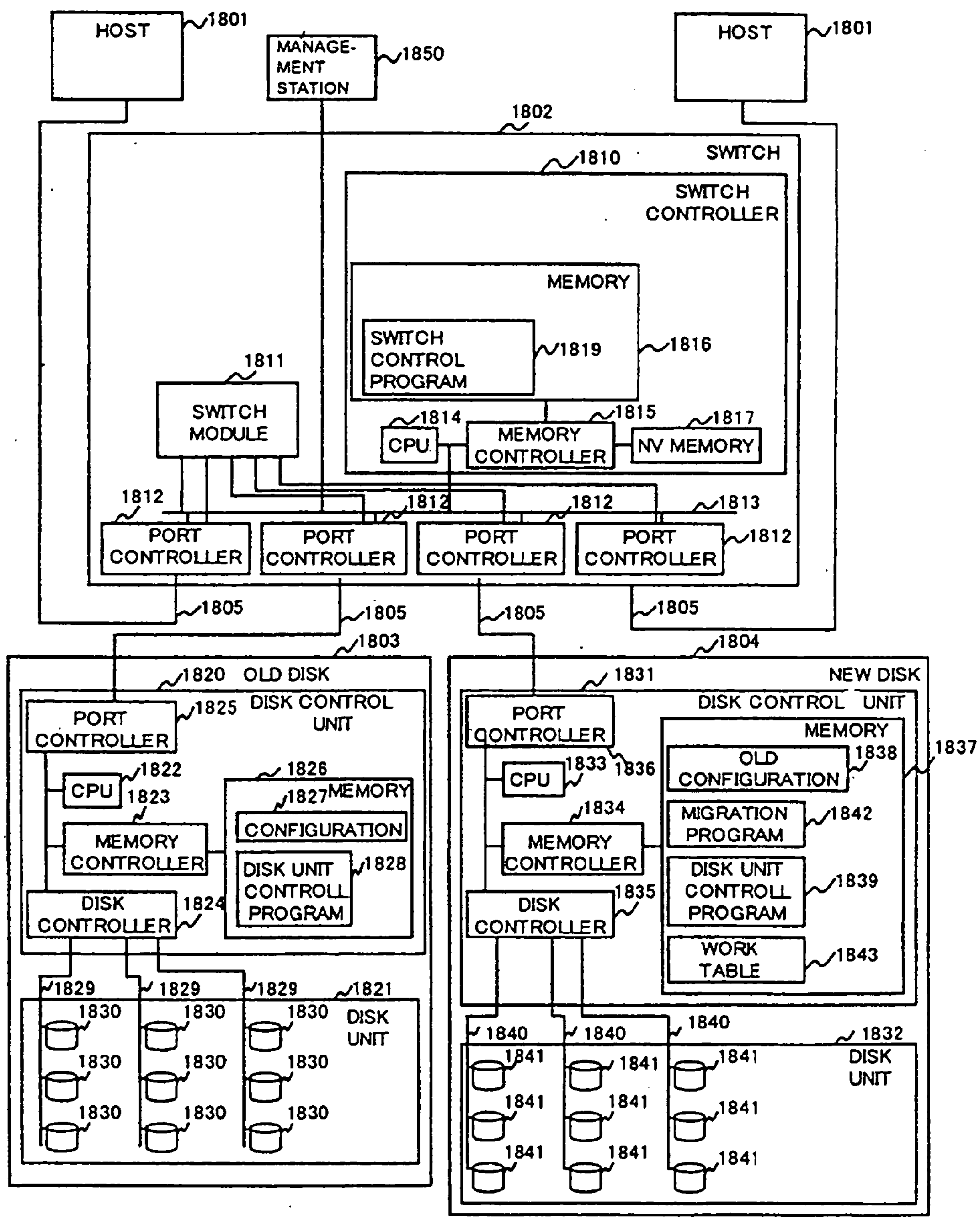
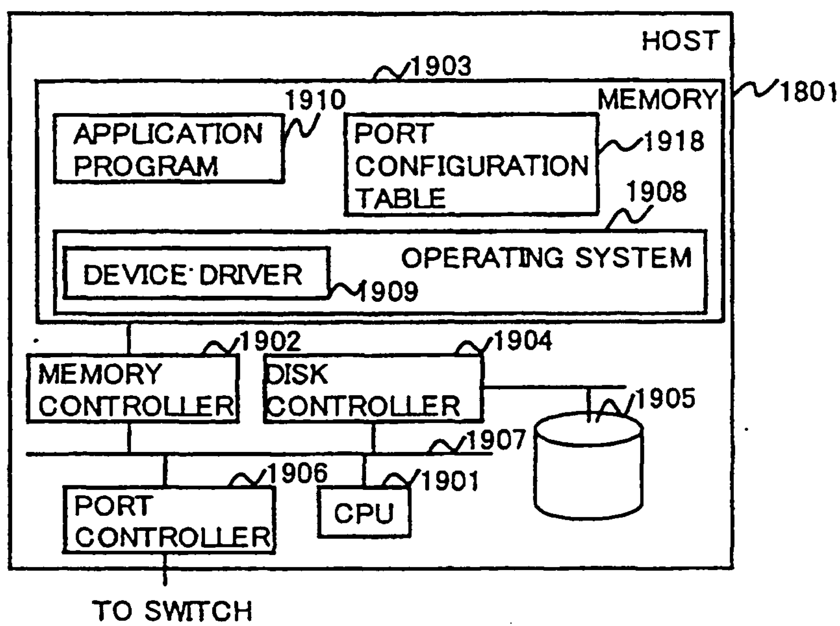


Fig 19


HOST
1903
1801
1910
MEMORY
APPLICATION PROGRAM
PORT CONFIGURATION TABLE
1918
1908
OPERATING SYSTEM
DEVICE DRIVER
1909
1902
1904
MEMORY CONTROLLER
DISK CONTROLLER
1905
1907
1906
1901
PORT CONTROLLER
CPU
TO SWITCH

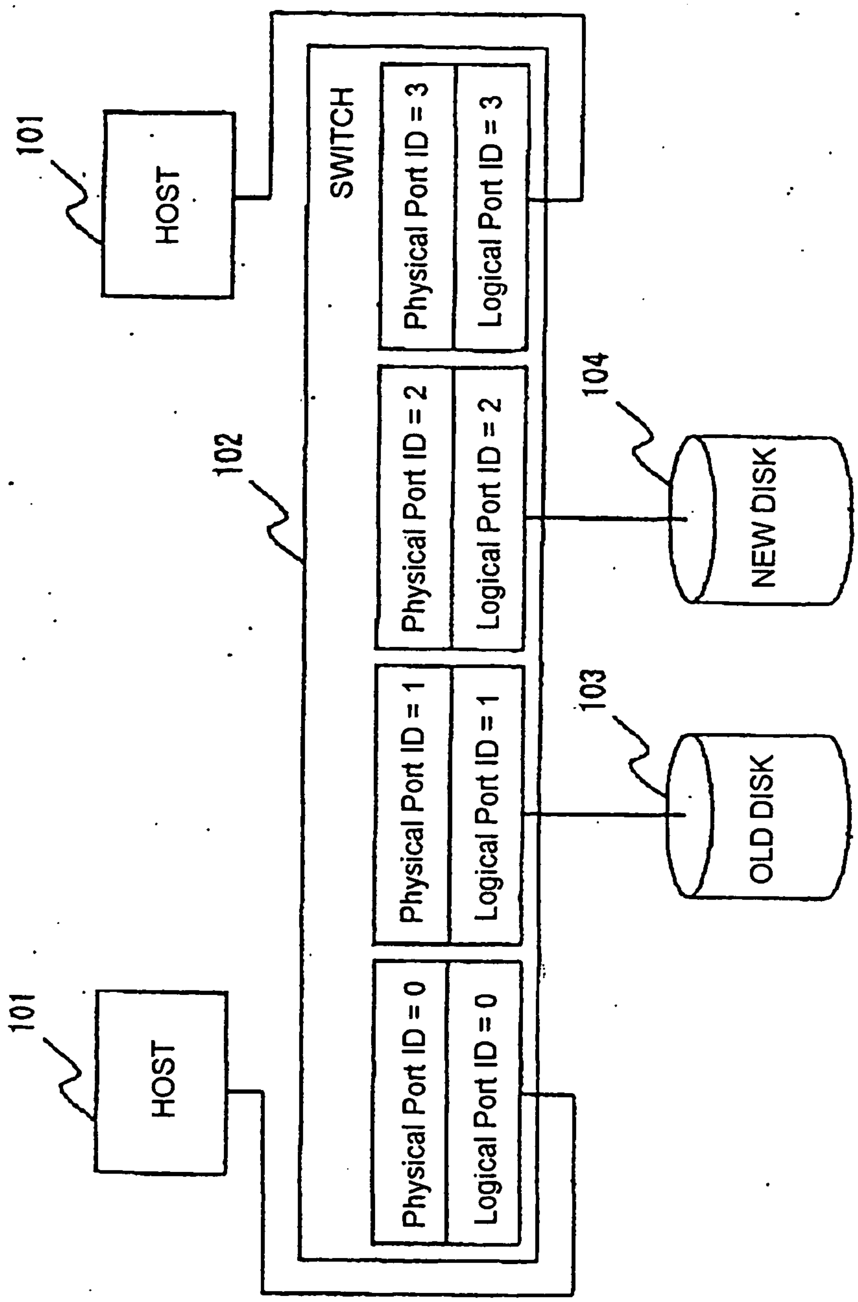

101
HOST
101
HOST
102
SWITCH
Physical Port ID = 0
Logical Port ID = 0
Physical Port ID = 1
Logical Port ID = 1
Physical Port ID = 2
Logical Port ID = 2
Physical Port ID = 3
Logical Port ID = 3
103
OLD DISK
104
NEW DISK

Fig 20

Fig 21

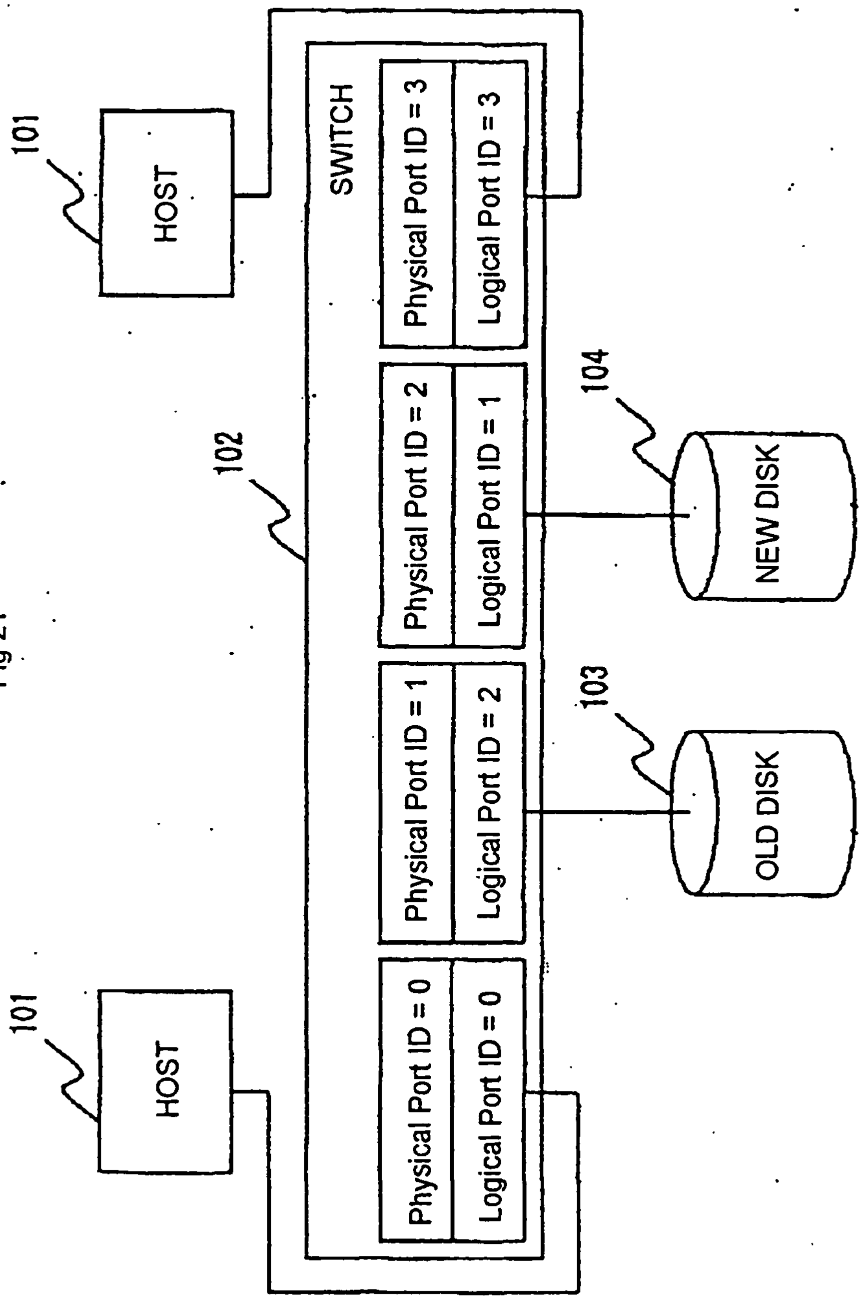
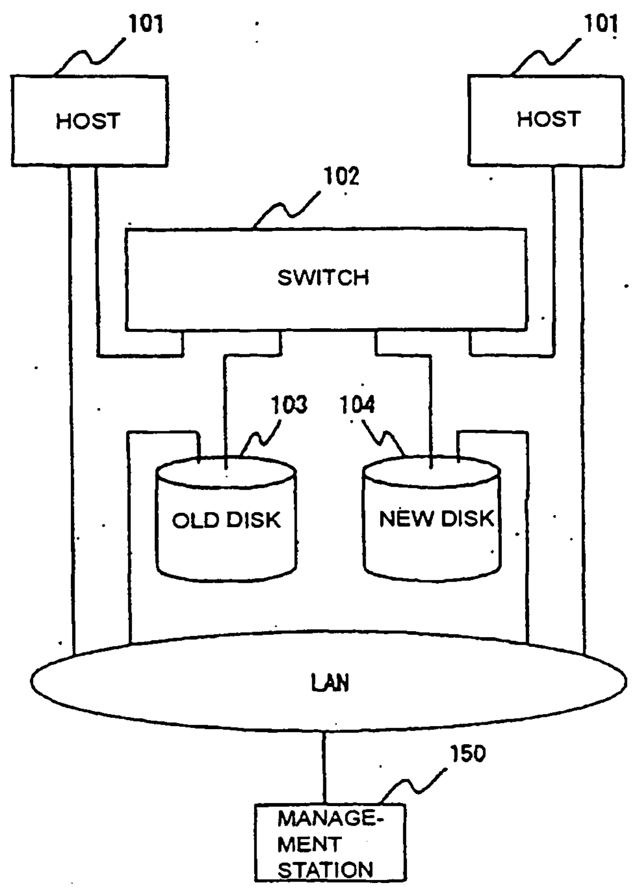

101
HOST
101
HOST
SWITCH
102
Physical Port ID = 0
Logical Port ID = 0
Physical Port ID = 1
Logical Port ID = 2
Physical Port ID = 2
Logical Port ID = 1
Physical Port ID = 3
Logical Port ID = 3
103
OLD DISK
104
NEW DISK

Fig 22


101
101
HOST
HOST
102
SWITCH
103
104
OLD DISK
NEW DISK
LAN
150
MANAGE-
MENT
STATION